

FOCUS for S/390

CA-IDMS Data Adapter User's Manual
Version 7.1

Contents

- 1 Introduction to the CA-IDMS Data Adapter1-1**
 - Ease of Use 1-2
 - Operating Environments..... 1-2
 - Efficiency 1-2
 - Security..... 1-3
 - Cross-Environment Access 1-3
 - Core Requirements 1-4
- 2 Getting Started2-1**
 - Using the CA-IDMS Data Adapter Under MVS2-2
 - Accessing Central Version and Local Mode2-2
 - Central Version Access in TSO.....2-2
 - Local Mode Access in TSO.....2-3
 - Interactive Access From MSO2-4
 - Central Version Batch Access2-4
 - Local Mode Batch Access.....2-5
 - Using the CA-IDMS Data Adapter Under CMS2-6
 - Accessing the CA-IDMS Data Adapter.....2-6
 - File Descriptions.....2-6
- 3 IDMS Overview and Mapping Concepts.....3-1**
 - Overview of IDMS3-2
 - Network Concepts3-2
 - Set-Based Relationships3-3
 - Simple Set3-4
 - Common Owner3-6
 - Common Member3-7
 - Multi-Member3-8
 - Bill-of-Materials.....3-9
 - Loop Structures3-12
 - CALC- and Index-Based Relationships3-13
 - Summary of Network Relationships3-14

4	CA-IDMS Data Adapter Master Files	4-1
	Accessing File Descriptions	4-2
	Master File Attributes	4-2
	File Attributes	4-3
	Segment Attributes	4-4
	Field Attributes	4-5
	The IDMS Database Key	4-8
	Remote Segment Descriptions	4-9
	Intra-Record Structures: The OCCURS Segment	4-9
	Describing Repeating Groups	4-10
	The POSITION Attribute	4-12
	The ORDER Field	4-13
5	CA-IDMS Data Adapter Access Files	5-1
	Access File Overview	5-2
	Access File Syntax	5-2
	Subschema Declaration Attributes	5-3
	Segment Attributes for Network Record-Types	5-5
	Index Declarations for Network Record-Types	5-9
	Dynamically Setting the DBNAME and DICTNAME	5-9
6	Creating File Descriptions With AUTOIDMS	6-1
	AUTOIDMS Prerequisites	6-2
	How to Execute AUTOIDMS	6-2
	The AUTOIDMS Main Menu	6-3
	The AUTOIDMS Root Record Selection Screen	6-8
	The AUTOIDMS Record/Set Selection Screen	6-10
	The Descriptions Generated by AUTOIDMS	6-12
	File and Segment Attributes Generated by AUTOIDMS	6-12
	Field Attributes Generated by AUTOIDMS	6-13
	Changes to the Descriptions Generated by AUTOIDMS	6-15
	Creating a Record of Master Files Generated by AUTOIDMS	6-15
	Search Order for the AUTOIDMS Parameter Log File (MVS only)	6-16
	How to Restart AUTOIDMS	6-17
	AUTOIDMS Sample Sessions	6-18
	Creating the EMPFULL Master and Access Files	6-18
	Creating the EMPFILE Master and Access Files	6-29

7	CA-IDMS Data Adapter Reporting Techniques.....	7-1
	The Retrieval Subtree	7-2
	Retrieval Sequence	7-3
	Retrieval Sequence With Unique Segments	7-4
	Effects of Screening Conditions on Retrieval	7-5
	Restricting the Number of Records Retrieved.....	7-7
	Screening Conditions With Unique Segments	7-8
	Retrieving Short Paths.....	7-8
	Short Paths in Unique Descendants.....	7-8
	Short Paths in Non-Unique Descendants.....	7-9
	Selective ALL Prefix.....	7-11
	File Inversion.....	7-12
	The JOIN Command.....	7-14
8	CA-IDMS Record Retrieval Intervals	8-1
	Entry Segment Retrieval of Network Records	8-2
	Retrieval by IDMS Database Key	8-2
	Retrieval by CALC Field	8-3
	Retrieval by IDMS Index	8-4
	Retrieval by Area Sweep.....	8-6
	Descendant Segment Retrieval of Network Records.....	8-6
	Set-Based Retrieval	8-7
	CALC-Based Retrieval.....	8-8
	Index-Based Retrieval	8-8
	Retrieving LRF-Based Records.....	8-9
A	Error Messages and Troubleshooting.....	A-1
	Accessing CA-IDMS Data Adapter Error Files	A-2
	Generating Error Messages Online.....	A-2
	Common Errors and Response Codes	A-2
	Common Errors Correctable by Database Administration	A-3
	Common Errors Correctable by Operations	A-4

B CA-IDMS Data Adapter Samples B-1

Schema Sample: EMPSCHM..... B-2

Network Subschema Sample: EMPSS01 B-11

Master File for Network Sample B-14

Access File for Network Sample B-19

Master File for LRF Sample..... B-20

Access File for LRF Sample..... B-21

Sample of a Partial LRF Record..... B-21

EMPFULL Master File..... B-22

EMPFULL Access File B-31

EMPFIL Master File B-33

EMPFIL Access File B-34

Sample CA-IDMS Report B-35

C Tracing CA-IDMS Data Adapter Processing..... C-1

Using Traces..... C-2

Allocating Traces C-2

Activating, Deactivating, and Querying Traces..... C-4

Sample Traces C-6

Glossary G-1

Index I-1

Cactus, EDA, FIDEL, FOCCALC, FOCUS, FOCUS Fusion, Information Builders, the Information Builders logo, SmartMode, SNAPpack, TableTalk, and Web390 are registered trademarks and Parlay, SiteAnalyzer, SmartMart, WebFOCUS, and WorldMART are trademarks of Information Builders, Inc.

Acrobat and Adobe are registered trademarks of Adobe Systems Incorporated.

NOMAD is a registered trademark of Aonix.

UniVerse is a registered trademark of Ardent Software, Inc.

IRMA is a trademark of Attachmate Corporation.

Baan is a registered trademark of Baan Company N.V.

SUPRA and TOTAL are registered trademarks of Cincom Systems, Inc.

Impromptu is a registered trademark of Cognos.

Alpha, DEC, DECnet, NonStop, and VAX are registered trademarks and Tru64, OpenVMS, and VMS are trademarks of Compaq Computer Corporation.

CA-ACF2, CA-Datcom, CA-IDMS, CA-Top Secret, and Ingres are registered trademarks of Computer Associates International, Inc.

MODEL 204 and M204 are registered trademarks of Computer Corporation of America.

Paradox is a registered trademark of Corel Corporation.

StorHouse is a registered trademark of FileTek, Inc.

HP MPE/iX is a registered trademark of Hewlett Packard Corporation.

Informix is a registered trademark of Informix Software, Inc.

Intel is a registered trademark of Intel Corporation.

ACF/VTAM, AIX, AS/400, CICS, DB2, DRDA, Distributed Relational Database Architecture, IBM, MQSeries, MVS/ESA, OS/2, OS/390, OS/400, RACF, RS/6000, S/390, VM/ESA, VSE/ESA and VTAM are registered trademarks and DB2/2, Hiperspace, IMS, MVS, QMF, SQL/DS, WebSphere, z/OS and z/VM are trademarks of International Business Machines Corporation.

INTERSOLVE, and Q+E are registered trademarks of INTERSOLVE.

Orbix is a registered trademark of Iona Technologies Inc.

Approach and DataLens are registered trademarks of Lotus Development Corporation.

ObjectView is a trademark of Matesys Corporation.

ActiveX, FrontPage, Microsoft, MS-DOS, PowerPoint, Visual Basic, Visual C++, Visual FoxPro, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Teradata is a registered trademark of NCR International, Inc.

Netscape, Netscape FastTrack Server, and Netscape Navigator are registered trademarks of Netscape Communications Corporation.

NetWare and Novell are registered trademarks of Novell, Inc.

CORBA is a trademark of Object Management Group, Inc.

Oracle is a registered trademark and Rdb is a trademark of Oracle Corporation.

PeopleSoft is a registered trademark of PeopleSoft, Inc.

INFOAccess is a trademark of Pioneer Systems, Inc.

Progress is a registered trademark of Progress Software Corporation.

Red Brick Warehouse is a trademark of Red Brick Systems.

SAP and SAP R/3 are registered trademarks and SAP Business Information Warehouse and SAP BW are trademarks of SAP AG.

Silverstream is a trademark of Silverstream Software.

ADABAS is a registered trademark of Software A.G.

CONNECT:Direct is a trademark of Sterling Commerce.

Java, JavaScript, NetDynamics, Solaris, SunOS, and iPlanet are trademarks of Sun Microsystems, Inc.

PowerBuilder and Sybase are registered trademarks and SQL Server is a trademark of Sybase, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

Allaire and JRun are trademarks of Allaire Corporation.

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2001, by Information Builders, Inc. All rights reserved. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Printed in the U.S.A.

Preface

This documentation describes how to use the CA-IDMS® Read Only Data Adapter in the MVS™ and VM/CMS operating environments. It is intended for FOCUS® users who need to retrieve information from CA-IDMS®/DB data sources. It is available in the following formats: printed manual and Adobe® Acrobat®. References to MVS apply to all supported releases of MVS, OS/390®, and z/OS™. References to VM, VM/CMS, or CMS apply to all supported releases of VM/CMS and z/VM™.

How This Manual Is Organized

This manual includes the following chapters:

Chapter/Appendix		Contents
1	<i>Introduction to the CA-IDMS Data Adapter</i>	Introduces the data adapter, its operating environments, and its core requirements.
2	<i>Getting Started</i>	Describes how to access the data adapter.
3	<i>IDMS Overview and Mapping Concepts</i>	Discusses IDMS data source structures and how they are represented as FOCUS structures.
4	<i>CA-IDMS Data Adapter Master Files</i>	Describes the elements in a Master File that describes network record-types and Logical Record Facility (LRF)-based records.
5	<i>CA-IDMS Data Adapter Access Files</i>	Describes the elements in an Access File for network record-types and LRF-based records.
6	<i>Creating File Descriptions With AUTOIDMS</i>	Describes how to use the AUTOIDMS facility to automatically create Master and Access files.
7	<i>CA-IDMS Data Adapter Reporting Techniques</i>	Describes advanced reporting topics, such as the retrieval sequence, retrieving short paths, file inversion, and the JOIN command.
8	<i>CA-IDMS Record Retrieval Intervals</i>	Describes internal calls that the data adapter generates to retrieve records.
A	<i>Error Messages and Troubleshooting</i>	Describes how to generate error messages online, how to access errors files. Lists common errors and the associated IDMS response codes.
B	<i>CA-IDMS Data Adapter Samples</i>	Contains sample schemas, subschemas, Master Files, and Access Files .
C	<i>Tracing CA-IDMS Data Adapter Processing</i>	Describes how to activate and deactivate data adapter traces.

This manual also contains a glossary of common terms you may need to know when using the data adapter.

Documentation Conventions

The following conventions apply throughout this manual:

Convention	Description
<code>THIS TYPEFACE or this typeface</code>	Denotes syntax that you must enter exactly as shown.
<i>this typeface</i>	Represents a placeholder (or variable) in syntax for a value that you or the system must supply.
<u>underscore</u>	Indicates a default setting.
<i>this typeface</i>	Represents a placeholder (or variable) in a text paragraph, indicates a cross-reference, or emphasizes an important term.
this typeface	Indicates buttons, menu items, and dialog box options you can click or select.
{ }	Indicates two choices from which you must choose one. You type one of these choices, not the braces.
[]	Indicates a group of optional parameters. None are required, but you may select one of them. Type only the information within the brackets, not the brackets.
	Separates two mutually exclusive choices in a syntax line. You type one of these choices, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameters, not the ellipsis points (...).
.	Indicates that there are (or could be) intervening or additional commands.

Related Publications

See the Information Builders® *Technical Publications Catalog* for the most up-to-date listing and prices of technical publications, plus ordering information. To obtain a catalog, contact the Publications Order Department at (800) 969-4636.

You can also visit our World Wide Web site, <http://www.informationbuilders.com>, to view a current listing of our publications and to place an order.

Customer Support

Do you have questions about the CA-IDMS Data Adapter?

Call Information Builders Customer Support Service (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your CA-IDMS Data Adapter questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code number (xxxx.xx) when you call.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our World Wide Web site, <http://www.informationbuilders.com>. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of www.informationbuilders.com also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Information You Should Have

To help our consultants answer your questions most effectively, be ready to provide the following information when you call:

- Your six-digit site code number (xxxx.xx).
- The FOCEXEC procedure (preferably with line numbers).
- Master File with picture (provided by CHECK FILE).
- Run sheet (beginning at login, including call to FOCUS), containing the following information:
 - ? RELEASE
 - ? FDT
 - ? LET
 - ? LOAD
 - ? COMBINE
 - ? JOIN
 - ? DEFINE
 - ? STAT
 - ? SET/? SET GRAPH
 - ? USE
 - ? TSO DDNAME or CMS QFI

- The exact nature of the problem:
 - Are the results or the format incorrect; are the text or calculations missing or misplaced?
 - The error message and code, if applicable.
 - Is this related to any other problem?
- Has the procedure or query ever worked in its present form? Has it been changed recently? How often does the problem occur?
- What release of the operating environment are you using? Has it, FOCUS, your security system, or a data adapter system changed?
- Is this problem reproducible? If so, how?
- Have you tried to reproduce your problem in the simplest form possible? For example, if you are having problems joining two data sources, have you tried executing a query containing just the code to access the data source?
- Do you have a trace file?
- How is the problem affecting your business? Is it halting development or production? Do you just have questions about functionality or documentation?

User Feedback

In an effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual. Please use the Reader Comments form at the end of this manual to relay suggestions for improving the publication or to alert us to corrections. You can also use the Document Enhancement Request Form on our Web site, <http://www.informationbuilders.com>.

Thank you, in advance, for your comments.

Information Builders Consulting and Training

Interested in training? Information Builders Education Department offers a wide variety of training courses for this and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our World Wide Web site (<http://www.informationbuilders.com>) or call (800) 969-INFO to speak to an Education Representative.

CHAPTER 1

Introduction to the CA-IDMS Data Adapter

Topics:

- Ease of Use
- Operating Environments
- Efficiency
- Security
- Cross-Environment Access
- Core Requirements

The CA-IDMS Data Adapter supports facilities for handling traditional network record-types, and Logical Records (LR).

Note:

The remainder of this manual will use the following terms interchangeably:

CA-IDMS data source IDMS data source

CA-IDMS Data Adapter IDMS Data Adapter

When end users issue FOCUS requests to access IDMS data sources, the IDMS Data Adapter generates the most efficient DML calls to satisfy this request. The IDMS Data Adapter retrieves the IDMS records, and FOCUS generates the reports. End users do not need to know Data Manipulation Language (DML) or how to navigate IDMS data sources.

The IDMS Data Adapter supports all FOCUS read-only facilities such as: reporting (TABLE, TABLEF, TABLETALK), creating graphs (GRAPH), statistical analysis (ANALYSE), interactive procedures (Dialogue Manager), and a system editor (TED). You can also report from multiple data sources (using MATCH FILE or JOIN) and create extract files. This is a read-only data adapter, therefore, write-only facilities such as MODIFY and FSCAN are not supported.

Ease of Use

To make an IDMS data source accessible to FOCUS, you need to describe the IDMS data source using FOCUS syntax. The FOCUS description is maintained in two files, a Master File and an associated Access File. FOCUS uses the Master File to interpret requests and to generate reports. The data adapter uses the Access File to make the DML and LRF calls that retrieve IDMS records. Use your system editor or the AUTOIDMS facility to create the file descriptions. You only need to create the file descriptions once for each IDMS subschema.

Operating Environments

The IDMS Data Adapter operates in conjunction with FOCUS in the MVS, OS/390, and VM/CMS operating environments. The data adapter can also run in a cross-machine environment: FOCUS on CMS can retrieve data from MVS, MSO, and VSE/ESA®.

The IDMS Data Adapter is entirely compatible with Computer Associates IDMS Release 12.0 and above.

Generally, the data adapter and FOCUS operate with IDMS data sources that reside in the same environment. In the MVS environment, IDMS data sources are accessed through the MVS version of FOCUS under TSO or in batch. In the CMS environment, the VM/CMS version of IDMS is used with the VM/CMS version of FOCUS. In this case, the IDMS data source consists of one or more CMS files that reside on a CMS minidisk. The CMS version of IDMS creates and maintains these CMS files. The object-time subschemas used are found in the CMS files.

- The data adapter can operate in IDMS local mode or Central Version (CV). IDMS determines the operation mode based on the existence (at execution time) of the IDMS Central Version System Control file (SYSCTL DDname) in MVS or based on the IDMSOPTI module parameters in CMS.
- The data adapter does not require language pre-compilers. Communication with IDMS software is established at run time. The data adapter dynamically generates calls to IDMS.

Efficiency

The IDMS Data Adapter is extremely efficient because:

1. IDMS data sources are not physically recreated as FOCUS data sources in order to make them intelligible to FOCUS. Only Master and Access Files are required to describe the IDMS data source to FOCUS.
2. End users can access IDMS data sources easily with FOCUS requests. No 3GL programming is required.

3. IDMS resources used to access records are kept to a minimum because the data adapter:
 - Reads only those IDMS areas needed for report generation.
 - Binds and accesses only the records needed to obtain report data.
 - Minimizes IDMS I/O by creating optimal DML and LRF calls based on the available selection criteria (IF and WHERE criteria).
 - Minimizes IDMS I/O based on a user-specified point of entry (alternate file view).

Security

The integrity of your IDMS data source is not jeopardized because FOCUS read-only features are supported.

The IDMS Data Adapter also supports FOCUS DBA security features. For example, you can use the ENCRYPT command on both Master and Access Files to prevent end users from examining sensitive data.

You can use FOCUS to enforce security at the following levels:

File level	Prevents access to a file.
Field level	Limits user access to fields within a file.
Field value	Controls access based on user-specified field values.
Resource security	Controls the number of records in a file that a user can retrieve for reporting.

Cross-Environment Access

A number of options are available to extend the range of FOCUS reporting in one environment to IDMS data sources that reside in another:

- You can access MVS, VM, or VSE/ESA IDMS data sources with FOCUS running under MVS or VM/CMS using Computer Associates' Distributed Database System (DDS) package. The data adapter uses the data source table name and DDS node name specified in the Access File to call the local IDMS CV to access a second operating system through DDS. No special FOCUS software is required.
- You can access MVS IDMS data sources from FOCUS running in a host VM environment on the same machine. This facility is available through Computer Associates' Universal Communications Facility (UCF). The use of this facility is transparent to FOCUS.

- You can access MVS or VSE/ESA IDMS data sources with FOCUS running under CMS using the FOCUS Cross Machine Interface (XMI). FOCUS residing in environment A with the IDMS Data Adapter issues a call to the XMI facility residing in environment B with the IDMS data sources. The IDMS/XMI facility issues standard calls to IDMS running in environment B without special Computer Associates software. Communication between the environments is aided by a virtual, real, or channel-to-channel adaptor dedicated to IDMS record-at-a-time retrieval.

In all possible scenarios, the Master and Access File and FOCEXEC procedures, must reside in the FOCUS environment.

Core Requirements

For MVS, the minimum virtual memory requirements are 4700K for FOCUS, 200K for the data adapter, and the appropriate requirements for IDMS and its I/O buffers.

For VM/CMS, the minimum virtual machine requirements are 2700K for FOCUS, 200K for the data adapter, and the appropriate requirements for IDMS and its I/O buffers.

Note:

It is recommended that 5000K be allocated for FOCUS in both MVS and VM/CMS.

Extra memory benefits FOCUS in that larger sorted reports can be generated entirely in memory without intermediate merge phases. The extra memory, however, has no impact on the retrieval efficiency.

Since IDMS treats FOCUS as an application program, all IDMS components, such as data sources, object-time subschemas, and executable IDMS software, must be specified for FOCUS to use.

CHAPTER 2

Getting Started

Topics:

- Using the CA-IDMS Data Adapter Under MVS
- Using the CA-IDMS Data Adapter Under CMS
- Accessing the CA-IDMS Data Adapter
- File Descriptions

The following topics contain procedures for running the CA-IDMS Data Adapter in the following environments:

- MVS in both interactive and batch mode.
- MVS in the MSO environment.
- CMS environment.

Using the CA-IDMS Data Adapter Under MVS

The data adapter is provided on your FOCUS distribution tape. The files necessary to make the data adapter operative will automatically be placed on the FOCUS production disk by following the standard FOCUS installation instructions.

Instructions for accessing IDMS data sources differ for Local Mode access and Central Version (CV) access.

Accessing Central Version and Local Mode

The IDMS load modules, IDMS and IDMSINTB, must be made available at run-time and assigned to ddname STEPLIB.

When using a CLIST, the STEPLIB ddname is not valid. You can either assign these load module members to the link list or you can allocate them in the TSO logon procedure. Contact your systems programmer to add these members.

The member names of the FOCUS Master and Access Files to read the subschema must be identically named and made available at run time.

The SYSIDMS ddname should be allocated to identify the DMCL.

Central Version Access in TSO

The ddname SYSCTL must be allocated to the SYSCTL data set corresponding to the Central Version desired. No allocation of IDMS data sources or subschema/DMCL load modules is needed when running CV.

The IDMS functions will take place in the IDMS CV address space. The subschema load modules can reside in either the dictionary load area (DDLDCLOD) or a load library. If the load modules reside in both places, IDMS uses the first one it finds based on the IDMS LOADLIST and dictionary established for your session. Contact your IDMS database administrator for additional assistance.

Use the following CLIST as a model for accessing FOCUS in Central Version:

```
ALLOC F(FOCEXEC) DA('prefix.FOCEXEC.DATA') SHR REUSE
ALLOC F(MASTER) DA('prefix.MASTER.DATA') SHR REUSE
ALLOC F(FOCIDMS) DA('prefix.ACCESS.DATA') SHR REUSE
ALLOC F(USERLIB) DA('prefix.IDMS.LOAD' -
                    'prefix.FUSELIB.LOAD') SHR REUSE
ALLOC F(FOCLIB) DA('prefix.FOCLIB.LOAD') SHR REUSE
ALLOC F(ERRORS) DA('prefix.IDMS.DATA' -
                    'prefix.ERRORS.DATA') SHR REUSE
ALLOC F(SYSCTL) DA('highlvl.SYSCTL') SHR REUSE
ALLOC F(SYSIDMS) DA('highlvl.SYSIDMS') SHR REUSE
CALL 'prefix.FOCLIB.LOAD(FOCUS)'
```


where:

highlvl

Is the high-level qualifier for Computer Associates supplied data sets.

prefix

Is the high-level qualifier for FOCUS production data sets.

Local Mode Access in TSO

The user must allocate all IDMS data sources. These data sources must be assigned to the ddnames that are assigned in the IDMS application schema/segment. Note that when you access IDMS in local mode from a CLIST, you have batch access to the data sources. Updates to the data that occur after you issue a retrieval request will not be reflected in your answer set until you issue another retrieval request.

All journal files and the default local mode tape journal file, SYSJRNL, must be specified and assigned to DD DUMMY.

When running the data adapter in a batch job or from an MSO server, the load modules must be allocated to ddname STEPLIB.

Use the following CLIST as a model for accessing FOCUS in local mode:

```
ALLOC F(FOCEXEC) DA('prefix.FOCEXEC.DATA') SHR REUSE
ALLOC F(MASTER) DA('prefix.MASTER.DATA') SHR REUSE
ALLOC F(FOCIDMS) DA('prefix.ACCESS.DATA') SHR REUSE
ALLOC F(USERLIB) DA('prefix.IDMS.LOAD' -
                    'prefix.FUSELIB.LOAD') SHR REUSE
ALLOC F(FOCLIB) DA('prefix.FOCLIB.LOAD') SHR REUSE
ALLOC F(ERRORS) DA('prefix.IDMS.DATA' -
                    'prefix.ERRORS.DATA') SHR REUSE
ALLOC F(SYSJRNL) DUMMY
ALLOC F(SYSIDMS) DA('highlvl.SYSIDMS') SHR REUSE
ALLOC F(FILE1) DA('user.IDMS.FILE1') SHR REUSE
ALLOC F(FILE2) DA('user.IDMS.FILE2') SHR REUSE
CALL 'prefix.FOCLIB.LOAD(FOCUS)'
```

where:

highlvl

Is the high-level qualifier for Computer Associates supplied data sets.

prefix

Is the high-level qualifier for FOCUS production data sets.

user

Is the high-level qualifier for a user's data source.

Interactive Access From MSO

For interactive access in MSO, modify the MSO startup JCL to include the following allocations. See the *Multi-Session Option Installation and Technical Reference Guide* for further information.

```
//STEPLIB DD DSN=highlvl.DBA.LOADLIB,DISP=SHR
//          DD DSN=prefix.IDMS.LOAD,DISP=SHR
//          DD DSN=highlvl.LOADLIB,DISP=SHR
//ERRORS   DD DSN=prefix.IDMS.DATA,DISP=SHR
//          DD DSN=prefix.ERRORS.DATA,DISP=SHR
//MASTER   DD DSN=prefix.MASTER.DATA,DISP=SHR
//FOCIDMS   DD DSN=prefix.ACCESS.DATA,DISP=SHR
//FOCEXEC   DD DSN=prefix.FOCEXEC.DATA,DISP=SHR
//SYSCTL    DD DSN=highlvl.SYSCTL,DISP=SHR
//SYSIDMS   DD DSN=highlvl.SYSIDMS,DISP=SHR
```

where:

highlvl

Is the high-level qualifier for Computer Associates supplied data sets.

prefix

Is the high-level qualifier for FOCUS production data sets.

Note:

In most cases there are two CA-IDMS load libraries that must be assigned to STEPLIB.

Central Version Batch Access

For batch access in Central Version, use the following JCL as a model:

```
//IDMSCV    EXEC PGM=FOCUS
//STEPLIB   DD DSN=highlvl.DBA.LOADLIB,DISP=SHR
//          DD DSN=highlvl.LOADLIB,DISP=SHR
//          DD DSN=prefix.IDMS.LOAD,DISP=SHR
//          DD DSN=prefix.FOCLIB.LOAD,DISP=SHR
//          DD DSN=prefix.FUSELIB.LOAD,DISP=SHR
//ERRORS    DD DSN=prefix.IDMS.DATA,DISP=SHR
//          DD DSN=prefix.ERRORS.DATA,DISP=SHR
//MASTER    DD DSN=prefix.MASTER.DATA,DISP=SHR
//FOCIDMS    DD DSN=prefix.ACCESS.DATA,DISP=SHR
//FOCEXEC    DD DSN=prefix.FOCEXEC.DATA,DISP=SHR
//SYSCTL     DD DSN=highlvl.SYSCTL,DISP=SHR
//SYSIDMS    DD DSN=highlvl.SYSIDMS,DISP=SHR
//SYSIN     DD *
? REL
TABLE FILE EMPFILE
PRINT EMP_NAME_0415
END
FIN
```

where:

highlvl

Is the high-level qualifier for Computer Associates supplied data sets.

prefix

Is the high-level qualifier for FOCUS production data sets.

In most cases there are two CA-IDMS load libraries that need to be assigned to STEPLIB.

Although in the sample JCL DDNAME SYSIN is allocated to in-stream input commands, it could have been allocated to a PDS member or sequential file.

Local Mode Batch Access

For batch access in Local Mode, use the following JCL as a model:

```
//IDMSLC      EXEC  PGM=FOCUS
//STEPLIB     DD   DSN=highlvl.DBA.LOADLIB,DISP=SHR
//            DD   DSN=highlvl.LOADLIB,DISP=SHR
//            DD   DSN=prefix.IDMS.LOAD,DISP=SHR
//            DD   DSN=prefix.FOCLIB.LOAD,DISP=SHR
//            DD   DSN=prefix.FUSELIB.LOAD,DISP=SHR
//ERRORS      DD   DSN=prefix.IDMS.DATA,DISP=SHR
//            DD   DSN=prefix.ERRORS.DATA,DISP=SHR
//MASTER      DD   DSN=prefix.MASTER.DATA,DISP=SHR
//FOCIDMS      DD   DSN=prefix.ACCESS.DATA,DISP=SHR
//FOCEXEC      DD   DSN=prefix.FOCEXEC.DATA,DISP=SHR
//SYSJRNL      DD   DUMMY
//SYSIDMS      DD   DSN=highlvl.SYSIDMS,DISP=SHR
//FILE1        DD   DSN=user.IDMS.FILE1,DISP=SHR
//FILE2        DD   DSN=user.IDMS.FILE2,DISP=SHR
//SYSIN        DD   *
? REL
TABLE FILE EMPFILE
PRINT EMP_NAME_0415
END
FIN
```

where:

highlvl

Is the high-level qualifier for Computer Associates supplied data sets.

prefix

Is the high-level qualifier for FOCUS production data sets.

In most cases there are two CA-IDMS load libraries that need to be assigned to STEPLIB.

Although in the sample JCL DDNAME SYSIN is allocated to in-stream input commands, it could have been allocated to a PDS member or sequential file.

The user must allocate all IDMS data sources. These data sources must be allocated to the ddnames that are assigned in the IDMS application schema/segment. All journal files and the default local mode tape journal file, SYSJRNL, must be allocated to DD DUMMY. There is no allocation for ddname SYSCTL in local mode.

Using the CA-IDMS Data Adapter Under CMS

The data adapter is provided on your FOCUS distribution tape. The files necessary to make the data adapter operative will automatically be placed on the FOCUS production disk by following the standard FOCUS installation instructions.

For local mode, Shared DASD, and UCF access, you must issue a FILEDEF command for each IDMS data source file. You may place these FILEDEF commands in the FOCUS EXEC, in a global FOCUS profile, a local user profile, a FOCEXEC, a CMS EXEC, or issue them from the command line. To access Central Version data sources, no special FILEDEF commands are required.

In addition, users will need to link to the Computer Associates production disk and issue the following GLOBAL commands:

```
GLOBAL TXTLIB IDMSLIB  
GLOBAL LOADLIB libname
```

where:

libname

Is the name of the load library into which IDMS was link edited.

To access IDMS data sources through the Cross Machine Interface, the GATEWAY id and the target job must be started.

Accessing the CA-IDMS Data Adapter

In the Master File, specify SUFFIX=IDMSR. The IDMSR module contains a new production version of the data adapter.

File Descriptions

The data adapter requires a Master and Access File for each IDMS data source referenced by FOCUS. The Master File is a layout of the records. The Access File contains specific IDMS data source information such as CALC keys and indexes. Refer to Chapter 6, *Creating File Descriptions With AUTOIDMS*, for more information about these descriptions.

CHAPTER 3

IDMS Overview and Mapping Concepts

Topics:

- Overview of IDMS
- Network Concepts

This chapter explains how IDMS data sources, both network and LRF-based, correspond to hierarchical and relational data sources. You should become familiar with this chapter, because most of its concepts affect the creation of Master and Access Files.

Overview of IDMS

IDMS is a network database management system that is accessed by a subschema (the equivalent in FOCUS is a Master File). IDMS provides two methods of retrieving records within a subschema from an application program: DML (network access) and LRF (LRF-based access).

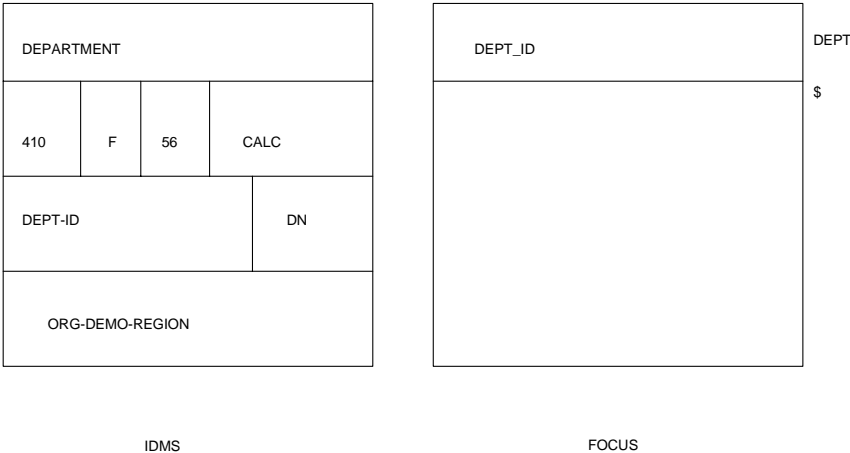
DML access is the traditional method of IDMS database navigation. It is the network navigation facility. Each physical record is retrieved separately. An application program enters the IDMS data source at a particular IDMS record-type (the equivalent in a Master File is a segment) and searches the set connections to retrieve the required data. See *Network Concepts* on page 3-2 for concepts that apply to network record-types.

Network Concepts

An IDMS record-type is described in a Master File as a segment. Since DML retrieval is record-oriented, not field-oriented, the Master File must list the fields in the same order as they appear on the IDMS record-type. However, a Master File does not have to list every field of a particular record-type; you can omit fields after a given field. For example, your description can list the first four fields of a 10-field IDMS record-type. (See Chapter 4, *CA-IDMS Data Adapter Master Files*, for dummy fields that can be substituted for omitted fields.)

IDMS field names can be used in your Master File if they are 12 characters or less; otherwise, you may assign similar or more meaningful names.

The following figure illustrates how to describe a record-type as a segment. The record-type DEPARTMENT contains the field-type DEPT-ID; its corresponding segment DEPT contains a field named DEPT_ID.

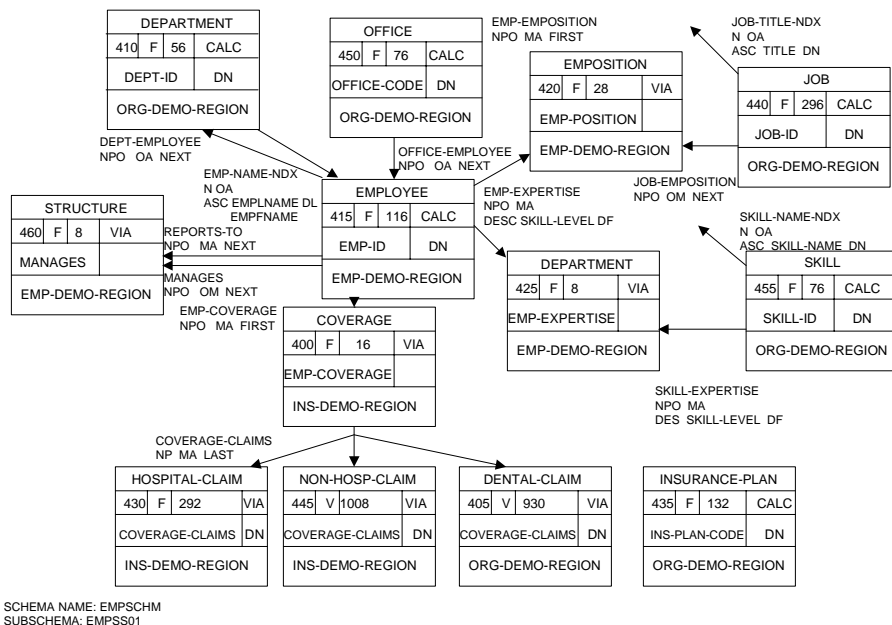


Repeating fields on an IDMS record-type are defined as OCCURS segments. For information on the OCCURS attribute, see Chapter 4, *CA-IDMS Data Adapter Master Files*.

Network record-types can be related to each other. These relationships can be physical (by set connections) or logical (achieved through an index or CALC field). The CA-IDMS Data Adapter supports both physical and logical relationships as illustrated in the following topics.

Set-Based Relationships

Within an IDMS data source, physical relationships between record-types are achieved with pointers that correspond to IDMS sets. A set implements a one-to-many relationship between record-types. (The Master File equivalent of a set is the parent/descendant relationship between segments.) In an IDMS set, one record-type acts as the owner (the one side of the relationship) and one or more record-types act as the members (the many side of the relationship). A single IDMS record-type can participate in several set relationships as either the owner or the member. Consider the following illustration of the EMPSS01 subschema:



In this example, the EMPLOYEE record-type participates as a member in the DEPT-EMPLOYEE and OFFICE-EMPLOYEE sets, with owners DEPARTMENT and OFFICE respectively. The EMPLOYEE record-type is also the owner in the EMP-EXPERTISE, EMP-EMPOSITION, and EMP-COVERAGE sets with members EXPERTISE, EMPOSITION, and COVERAGE.

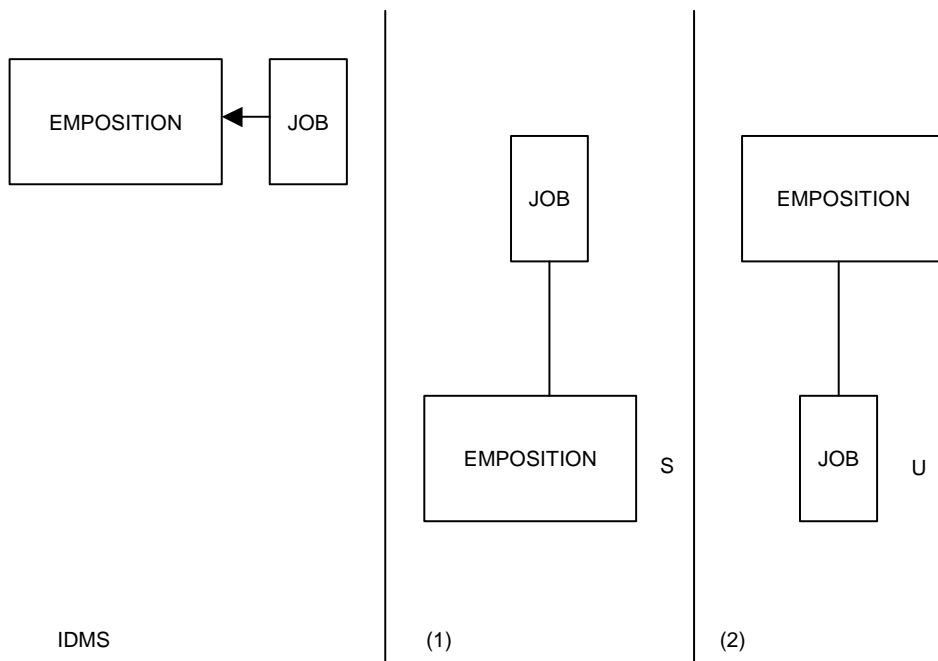
Many-to-many relationships using junction record-types also exist in IDMS data sources. In the example, EMPLOYEE is related to JOBS with EMPOSITION as the junction record-type, since an employee may have many jobs or a specific job can be held by more than one person. A junction record-type is a member in two or more sets with related owners.

The IDMS representation of record-types and set relationships within a data source is called an Entity Relationship diagram. In an Entity Relationship diagram, a record-type is depicted as a box and a set as a line with an arrow. Set names appear as labels beside the arrows. The box that the arrow points to is the member record-type. Triangles indicate indexes. The diagram in this topic is the Entity Relationship diagram for the IDMS network subschema EMPSS01. Sections of this diagram are referenced throughout the chapter (see Appendix B, *CA-IDMS Data Adapter Samples*, for its corresponding Master File EMPFULL).

Seven examples of set-based relationships are depicted. This manual refers to them as: simple set, common owner, common member, multi-member, bill-of-materials (simple and multi-tiered), and loop structures. All of these correspond to structures you can define in a Master File and are explained in subsequent topics.

Simple Set

The following diagram illustrates the basic mapping principle of a simple set: An IDMS record-type corresponds to a segment in a Master File; a set relationship corresponds to a parent/descendant relationship.



This figure also illustrates a second principle: an IDMS structure can have more than one representation as a hierarchy in a Master File. The type of parent/descendant relationship required in the Master File depends on whether the owner or the member record-type is designated as the parent segment.

The JOB-EMPOSITION set has two possible Master File representations:

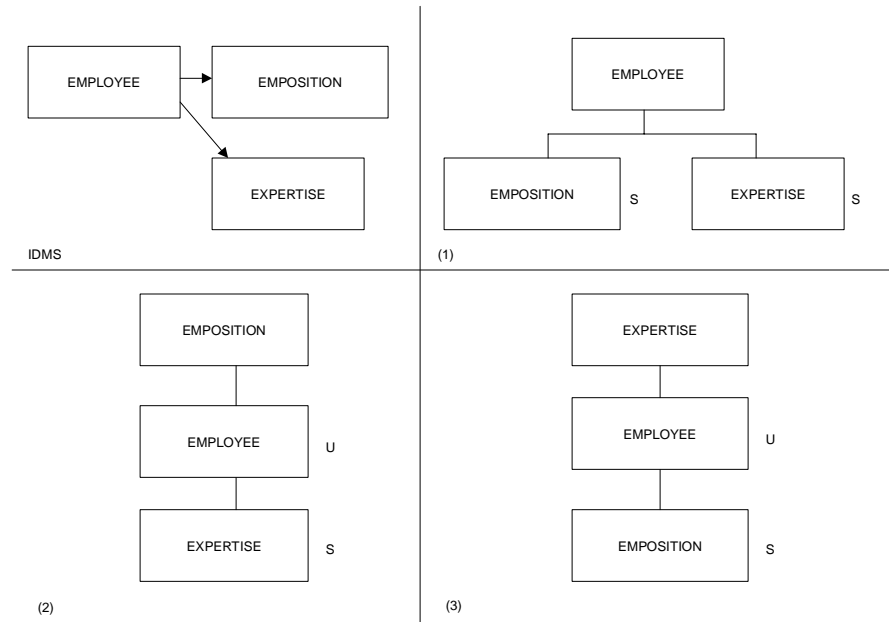
1. Shows the JOB record-type, the owner in the set, mapped as the root or entry segment. Since a member record-type is multiply occurring (for example, several instances of EMPOSITION records per JOB instance, indicating that many positions share one job title and description), the EMPOSITION record-type is displayed as a non-unique descendant.
2. Depicts the reverse. The EMPOSITION record-type is the root or entry segment and JOB is the unique descendant, since an EMPOSITION instance can have only one owner (for example, only one job title and description per position).

Note:

Panels 1 and 2 may be the results of a single Master File. Consult Chapter 7, *CA-IDMS Data Adapter Reporting Techniques*, for additional information.

Common Owner

Given the rules in *Simple Set* on page 3-4, consider a more complex scenario called a common owner. A common owner structure contains a record-type that is the owner of two or more record-types. Several representations are possible. For example, the following figure depicts three ways to describe the EMPLOYEE, EXPERTISE, and EMPOSITION structure in one Master File:

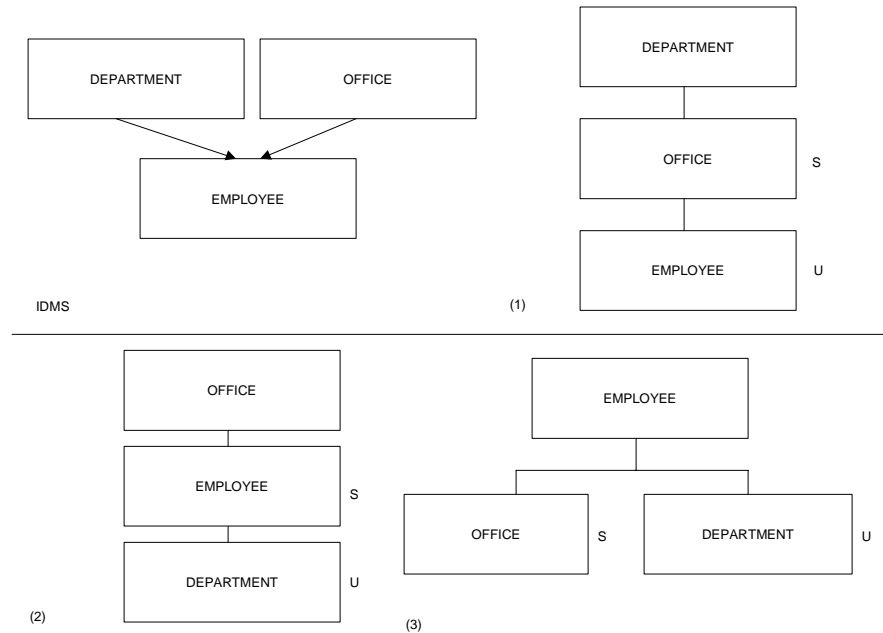


1. Shows the EMPLOYEE record-type, the owner in both sets, mapped as the root segment of EMPOSITION and EXPERTISE. Since an EMPLOYEE record can have many EMPOSITION and EXPERTISE records, both descendants are non-unique.
2. Depicts the EMPOSITION record-type as the root segment of EMPLOYEE, which acts as the parent of EXPERTISE. Since an EMPOSITION record can have only one owner, EMPLOYEE is a unique descendant; EXPERTISE is a non-unique descendant of EMPLOYEE.
3. Depicts the EXPERTISE record-type as the parent of EMPLOYEE, which acts as the parent of EMPOSITION. EMPLOYEE is a unique descendant and EMPOSITION is a non-unique descendant of EMPLOYEE.

Common Member

When an IDMS record-type is a member of two or more sets, the association of the owner record-type as the parent segment must be abandoned for one or more sets, because a segment in a Master File can have only one parent.

The following figure displays the possible interpretations for this IDMS configuration:



The EMPLOYEE record-type is a common member in the DEPT-EMPLOYEE and the OFFICE-EMPLOYEE sets. This structure can be described in a Master File in three ways:

1. Shows DEPARTMENT as the root segment with EMPLOYEE as its non-unique descendant; OFFICE is the unique descendant of EMPLOYEE.
2. Depicts the reverse: OFFICE is the root segment; EMPLOYEE is its non-unique descendant; DEPARTMENT is the unique descendant of EMPLOYEE.
3. Shows the only other alternative: EMPLOYEE is the parent of OFFICE and DEPARTMENT. Both descendants are unique, since an EMPLOYEE can belong to only one OFFICE and DEPARTMENT.

Notice that the rules for simple sets are still valid:

- If the owner record-type is the parent segment, the member record-type as a descendant segment is non-unique.
- If the member record-type is the parent segment, the owner record-type as a descendant segment is unique.
- A member or an owner record-type may act as a root segment. (It does not matter if the root segment is unique or non-unique.)

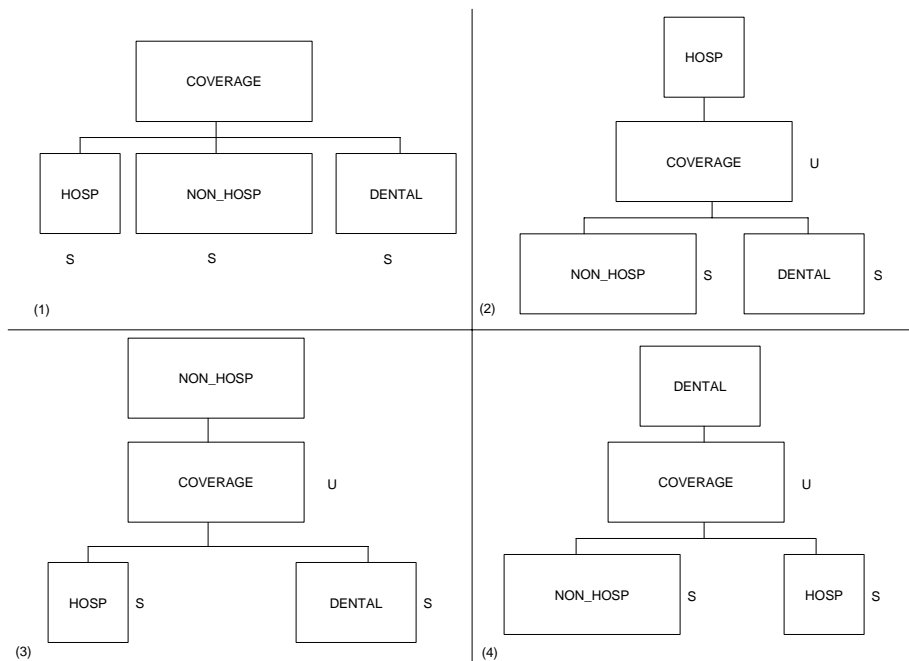
It may be helpful to think of the hierarchical depiction of a network structure as its navigational path. From an IDMS standpoint, panel 1 shows that the IDMS data source can be entered at the DEPARTMENT record-type. The corresponding EMPLOYEE record occurrences for a DEPARTMENT record occurrence can be obtained by searching the DEPT-EMPLOYEE set. For each EMPLOYEE record occurrence, the corresponding OFFICE record occurrence is retrieved by obtaining the owner in the OFFICE-EMPLOYEE set. This is a three-segment retrieval hierarchy that maps to Master Files.

Multi-Member

When there is more than one member record-type, the set is called a multi-member set.

A multi-member set is represented in the Master File exactly like a *Common Owner* on page 3-6. The fact that the two relationships are based on the same set is stated in the Access File.

For example, to describe the COVERAGE record-type and its three members of the COVERAGE-CLAIMS set, you may choose one of four ways as depicted in the following figure:



1. Shows the COVERAGE record-type, owner of the multi-member set, as the root segment. Since several instances of CLAIMS can be reported against one insurance policy (COVERAGE), each member is a non-unique descendant.

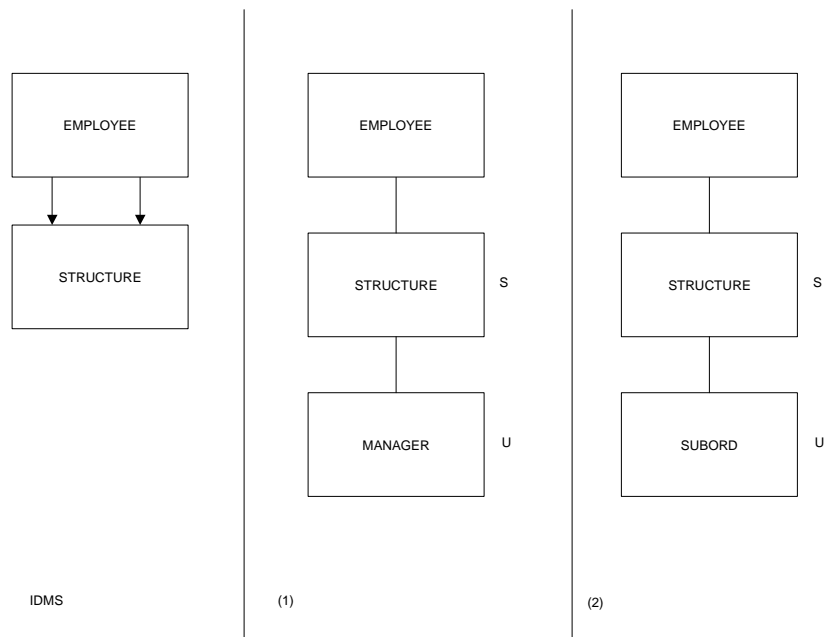
2. Depicts HOSPITAL-CLAIM as the parent of COVERAGE, and the other two member record-types as descendants of COVERAGE. In each case, a claim can be reported against only one insurance policy. This explanation applies to panels 3 and 4 as well.

Bill-of-Materials

Bill-of-materials structures are classified as simple or multi-tiered. In this topic, the simple version is discussed first.

Two record-types linked by more than one set are called a bill-of-materials structure. This structure describes a many-to-many relationship between record occurrences of the same record-type. The member record-type is the junction record-type between the two related owners.

For a simple bill-of-materials structure, the owner record-type must be represented as two or more segments in the Master File with different field names for the identical fields. This ensures that, at retrieval time, the field names you specify in a request will provide the proper navigational path. The following figure illustrates this type of structure:



The EMPLOYEE and STRUCTURE record-types are connected by two sets. This simple bill-of-materials structure can be described two ways:

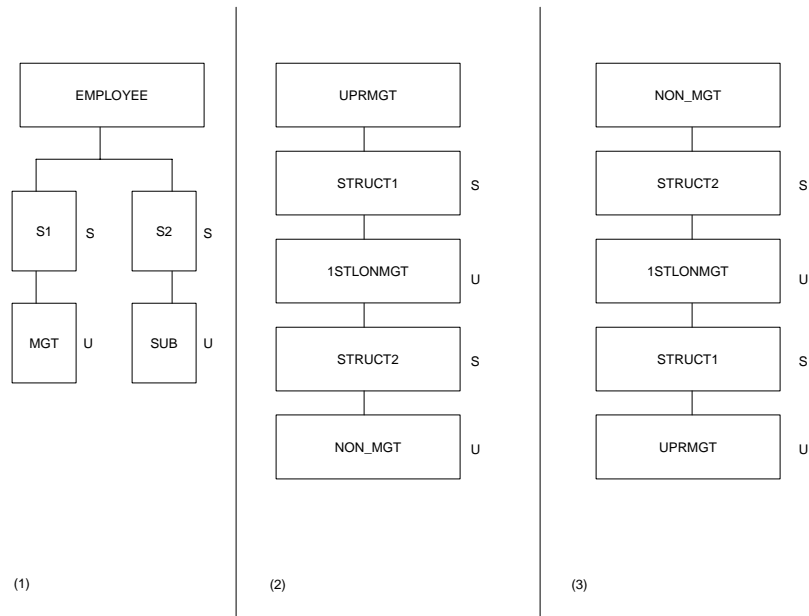
1. Shows the relationship as employee-to-manager. The EMPLOYEE record-type is the parent segment of the non-unique STRUCTURE segment using the REPORTS-TO set. The STRUCTURE record-type, in turn, is the parent segment of the unique MANAGER segment using the MANAGES set. (The MANAGER segment duplicates the EMPLOYEE segment, and its fields are renamed.)
2. Shows the relationship as employee-to-subordinate. The EMPLOYEE record-type is the parent segment of the non-unique STRUCTURE segment using the MANAGES set. The STRUCTURE record-type is the parent segment of the unique SUBORD segment using the REPORTS-TO set. (The SUBORD segment duplicates the EMPLOYEE segment and its fields are renamed.)

This structure represents a two-tiered employee-to-employee relationship. Multi-tiered relationships are extended bill-of-materials structures. Multi-tiered relationships are created and used for different levels of reporting. The number of levels or tiers should be kept to a minimum.

To determine the number of segments required to describe an n-tiered relationship, use this formula:

`Number of segments = (2 x number of tiers) - 1`

The following represents a three-tiered version of the previous structure:

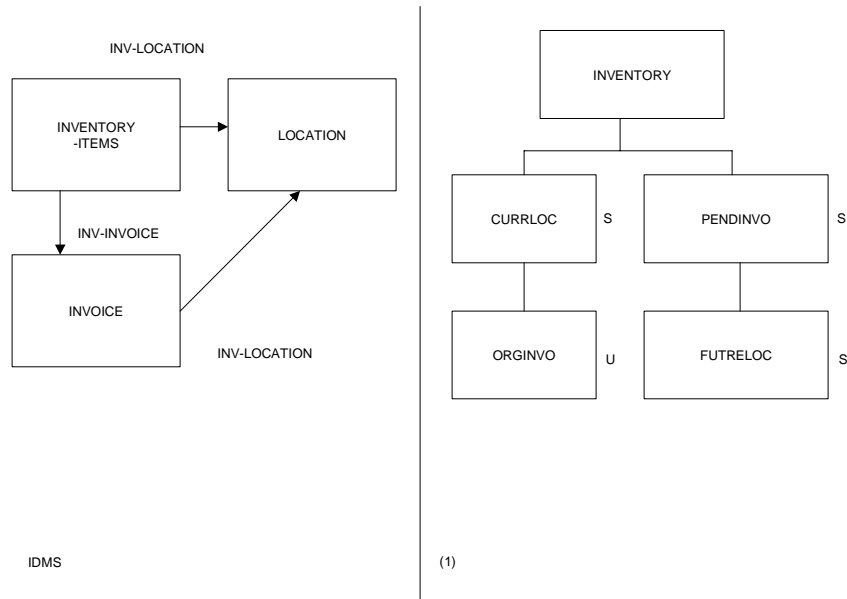


1. Combines both views with EMPLOYEE as the parent of two non-unique descendants, S1 and S2. Both S1 and S2, in turn, are parents of unique descendants, MGT and SUB, respectively. (S1 and S2 describe junction records that point to MGT and SUB.)
2. Shows a three-tiered relationship between employees implemented in a five-segment single-path hierarchy. The segments UPRMGT (upper management), 1STLONMGT (first-line management), and NON_MGT (non-management) all describe the EMPLOYEE segment but have renamed fields. (Like S1 and S2 above, STRUCT1 and STRUCT2 contain renamed fields that point to descendant segments.)
3. Depicts the opposite of panel 2.

Loop Structures

Loop structures in IDMS implement complicated relationships between record-types. For the Master File depiction of a loop, you must select a record-type to be the parent in the relationship.

Consider the following loop structure:



An **INVOICE** record occurrence has an owner record occurrence (**INVENTORY-ITEMS**) only if the invoice order is pending or has not been delivered. The **INVO-LOCATION** set lists the location where an invoice item will be or was delivered.

Panel 1 translates this structure as a multi-tiered structure with one tier of renamed segments. The **INVENTORY-ITEMS** record-type, in this case, acts as the root **INVENTORY**. The **LOCATION** record-type is mapped as the **CURRLOC** segment, which lists the location of items currently in stock. The **INVOICE** record-type is mapped as the **PENDINVO** segment, which lists pending invoice orders. Segments **ORGINVO** and **FUTRELOC** are required segments that rename the data in record-types **INVOICE** and **LOCATION**. The segment **ORGINVO** contains historical information for an inventory item. The segment **FUTRELOC** indicates where an ordered item will be delivered.

CALC- and Index-Based Relationships

Logical relationships, unlike physical ones, are based on the occurrence of the same data value in two different record-types. To make the parent/descendant connection, the IDMS Data Adapter uses CALC fields or indexes to locate the related record occurrences. The related fields are not required to have the same name in both record-types, but the field format must be the same. See Chapter 4, *CA-IDMS Data Adapter Master Files*, for additional information on format conversions.

- Selection criteria on CALC fields and indexes are also used to generate CALC and index calls to IDMS. This topic is discussed in detail in Chapter 8, *CA-IDMS Record Retrieval Intervals*.
- CALC or index fields can also be groups consisting of fields contiguous in both parent and descendant segments. The format types and lengths of the GROUP and its fields must be comparable in both parent and descendant segments. Consult Chapter 4, *CA-IDMS Data Adapter Master Files*, for more information.

The IDMS Data Adapter currently supports IDMS Integrated Indexes. As of IDMS Release 12, indexes can be integrated with the database management system (DBMS). The DBMS handles all index access and maintenance for Integrated Indexes.

Like set-based descendants, CALC- and index-based descendants are unique or non-unique, but this depends largely on how the DUPLICATES attribute is specified in the Access File. (The SEGTYPE attribute for the descendant must also reflect the DUPLICATES attribute; for example, if CLCDUP=N, SEGTYPE=U.) Unique descendants are treated in the same manner regardless of what underlies the parent/descendant relationship: set, index, or CALC field.

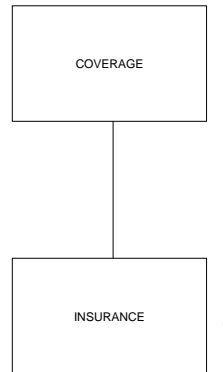
Note:

Inversions that reverse CALC- or index-based segments result in an error message if the segments are referenced in a report request.

The COVERAGE and INSURANCE-PLAN record-types in *Set-Based Relationships* on page 3-3 illustrate a logical relationship. The field PLAN-CODE is common to both record-types. The parent/descendant relationship can be described in the Master File if:

- The INSURANCE-PLAN record-type has an index on PLAN-CODE.
- The INSURANCE-PLAN record-type is an IDMS CALC record-type with PLAN-CODE as the CALC key.

The following figure depicts the Master File representation of the set-based relationship using the CALC field method. COVERAGE is the parent segment and INSURANCE-PLAN is the descendant. Since the DUPLICATES parameter is set to N (CLCDUP=N), the INSURANCE segment is unique:



Summary of Network Relationships

As illustrated in previous examples, there are four ways to implement a network relationship. Each can be described as a parent/descendant relationship:

- Owner/member
- Member/owner
- Field or GROUP/CALC field
- Field or Integrated Index

All four can be intermixed in one Master File. The actual underlying connection (set, CALC, index) between parent and descendant is not apparent to end users who only need to know Master File field names. In addition, a Master File can list record-types from multiple subschemas, data sources, and dictionaries.

It is important to remember that your Master File is a default description. You can manipulate your Master File at execution time through file inversion. File inversions and their implications are described in Chapter 7, *CA-IDMS Data Adapter Reporting Techniques*.

In report processing, the identity of the record-type behind a segment is invisible. The retrieval technique is followed in all cases as if all segments were represented by distinct records. Within IDMS, currencies are maintained by storing the DBKEYs of record-types and retrieving the record occurrence again, when needed.

The top-to-bottom order in which segments are defined, the chains of parent/descendant relationships, corresponds to structural relationships among the IDMS record-types. This top-to-bottom order is logically significant; the left-to-right order, on the other hand, is not.

CHAPTER 4

CA-IDMS Data Adapter Master Files

Topics:

- Accessing File Descriptions
- Master File Attributes
- Remote Segment Descriptions
- Intra-Record Structures: The OCCURS Segment

The description of an IDMS data source is contained in a pair of simple comma-delimited sequential files: a Master File, and an associated Access File. The Master File defines the field names for IDMS fields and segment relationships. The Access File is an extension of the Master File that provides additional information needed to retrieve data from IDMS data sources. The Access File provides information needed to map DML record-types and LRF-based records, including record and area names and, where needed, set or field name information.

Accessing File Descriptions

Master and Access Files are stored as comma-delimited sequential files. On VM, Master Files have filetype MASTER and Access Files have filetype FOCIDMS. On MVS, Master Files are members of a PDS allocated to ddname MASTER and Access Files are members of a PDS allocated to ddname FOCIDMS. Each Master File has a corresponding Access File. On VM, they have the same file names. On MVS, they have the same member names.

	VM/CMS	MVS
Master File	SALES MASTER	//MASTER DD DSN= <i>dsname1</i>
Access File	SALES FOCIDMS	//FOCIDMS DD DSN= <i>dsname2</i>

Note:

- CMS files can be on any accessed disk, not necessarily the same for both. The file name chosen for the Master File definition of an IDMS structure is purely suggestive and bears no relation to any internal IDMS names.
- MVS files are members of partitioned data sets allocated to the ddnames in the table. Associated Master and Access Files must have the same member names, SALES in this example. Any data set name is valid, but names that indicate the function of the data set are recommended for documentation purposes.

This chapter concentrates on the Master File and its components. It contains brief examples of syntax. (For more detailed examples, see Appendix B, *CA-IDMS Data Adapter Samples*.) Consult Chapter 5, *CA-IDMS Data Adapter Access Files*, for additional information.

Master File Attributes

A Master File consists of file, segment, and field declarations. Each declaration consists of some or all of the following attributes:

```
FILE=filename, SUFFIX=IDMSR,$
SEGNAME=segname, PARENT=parent
      SEGTYPE=segtype, CRFILE=crfile, OCCURS=occurs, POSITION=position, $
{GROUP=group|FIELDNAME=fieldname}, ALIAS=alias, USAGE=usage,
ACTUAL=actual,      FIELDTYPE=fieldtype, $
```

where:

filename
Is the Master File name.

IDMSR
Indicates that this Master File describes an IDMS data source.

segname

Is the segment name; eight characters maximum.

parent

Is the name of the parent segment.

segtype

Defines whether the segment is unique, U or KLU, or non-unique, S or KL. KLU and KL apply to segments that are described in a remote Master File (see *Remote Segment Descriptions* on page 4-9 for additional information).

crfile

Is the name of the remote Master File when SEGTYPE is KLU or KL.

occurs

Is the integer or field name that specifies the number of repeating fields.

position

Is the field name of the beginning OCCURS segment.

group

Is any name; 66 characters maximum.

fieldname

Is any name; 66 characters maximum.

alias

Is any name, 66 characters maximum; for an LRF last field, include .END suffix. For index, calc and key fields, either the field name or the alias must be a maximum of 12 characters.

usage

Is the usage format.

actual

Is the actual IDMS data format.

fieldtype

Is I if field has an IDMS Integrated Index. N/A for LRF records.

Note:

Attributes and their values are separated by commas; each declaration must end with a comma and dollar sign.

File Attributes

The FILE attribute names the Master File. In MVS, it is the member name within a partitioned data set (PDS) allocated to ddname MASTER. In CMS, it is the name of a file with file type MASTER.

The value of the SUFFIX attribute is IDMSR, the load module name of the CA-IDMS Data Adapter module.

Segment Attributes

You can assign any SEGNAME value to DML record-types and LRF records, not necessarily the IDMS record names. However, using a similar name is recommended for documentation purposes. The segment name can be a maximum of eight characters and must be unique within a given Master File. If the same IDMS record-type is viewed in two different contexts (for example, a loop structure), two segment declarations are required.

The PARENT attribute is required for descendant segment declarations. The PARENT value names the descendant's parent segment. This attribute is not specified for the root or entry segment declaration.

The SEGTYPE attribute indicates whether a segment can occur only once (SEGTYPE=U) or many times (SEGTYPE=S). It is used as follows:

- For root segments, SEGTYPE has no meaning and may be omitted entirely.
- For parent segments, SEGTYPE values can be S or U.
- For descendant segments with set-based relationships, SEGTYPE indicates whether the segment acts as an owner or a member. If the descendant segment is the owner record-type, the SEGTYPE value is U. If the descendant segment is the member, the SEGTYPE value is S.
- For descendant segments with CALC-based or index-based relationships, SEGTYPE values are S or U depending on the DUPLICATES (CLCDUP or IXDUP) value.
- For descendant segments with LRF-based relationships, SEGTYPE values may be S or U.

Note:

To the CA-IDMS Data Adapter, there is no functional difference between the specification of S, S0, S1, or Sn for the value of SEGTYPE.

If an inverted file view is used, SEGTYPE is reinterpreted to reflect the new situation.

The CRFILE attribute is specified only for segments that are described remotely in another Master File. The field descriptions for these segments are, in effect, copied into the Master File at execution time. Remote descriptions are discussed in *Remote Segment Descriptions* on page 4-9.

The OCCURS and POSITION attributes are specified only when the segment corresponds to an intra-record structure, such as a COBOL OCCURS or OCCURS DEPENDING structure. These attributes are described in detail in *Intra-Record Structures: The OCCURS Segment* on page 4-9.

Field Attributes

The GROUP attribute identifies a set of fields with a single name. The individual fields that comprise the group must be specified immediately following the group declaration. The GROUP name can be any unique name up to 66 characters in length. Its usage is similar to that of a COBOL group name. Generally, this attribute is used for IDMS indexed or CALC fields.

Note:

USAGE and ACTUAL format types for a GROUP field are always alphanumeric (A). USAGE and ACTUAL format lengths for a GROUP field are the sums of the field lengths that form the GROUP field.

The FIELDNAME attribute can be any name, not necessarily an IDMS field name, or it can remain blank. It can be up to 66 characters long. However, for index, calc, and key fields, either the field name or alias must be a maximum of 12 characters. The name that you assign must be unique within a segment because data is referenced through field names or aliases. Hyphenated names must be enclosed in single quotation marks. If the same IDMS record-type is viewed in different contexts and, therefore, underlies two or more segments, different field names must be specified in the separate segment descriptions.

Because of the record-oriented nature of IDMS, fields in the Master File are described for each segment in the same order as they appear in the subschema record area. (See Appendix B, *CA-IDMS Data Adapter Samples*, for a procedure that prints a report listing IDMS fields.) Bytes skipped for field alignment or for fields you want to omit must be described as dummy fields of the appropriate length. Dummy fields can have blank field names and aliases.

You do not need to describe all fields of the record-type or LRF record, only an initial set, starting with the first field and continuing up to the last field of interest. For variable-length record-types, treat the fixed-length portion as one segment and the variable portion as another segment, as described with the OCCURS attribute. See *Intra-Record Structures: The OCCURS Segment* on page 4-9 more information.

The ALIAS attribute specifies an alternate name for FIELDNAME. An alias name can also be up to 66 characters and must be unique within a segment. Certain fields require specific ALIAS values:

- Database key fields - the ALIAS value is DBKEY (see *The IDMS Database Key* on page 4-8).
- ORDER fields - the ALIAS value is ORDER (see *The ORDER Field* on page 4-13).
- Fields on LRF-based records (LR) that correspond to the last fields of their underlying physical record-types. Filler fields may be required if the last field does not end on a double-word boundary. The ALIAS for each filler field is a unique name with a maximum of eight characters counting the .END suffix. The data adapter uses this suffix to correctly address LRF records for LRF calls.
- GROUP fields - an ALIAS is always required or an error message results.

The USAGE and ACTUAL attributes describe the data format of the field.

The FIELDTYPE attribute identifies Integrated Index fields on network record-types. Omit it for LRF records.

Reference

Field Format Conversion

The ACTUAL attribute defines the field length for COBOL fields found in the IDMS file. The number of internal storage bytes used by COBOL determines the field's actual length for these formats:

- Alphanumeric (A) The number of characters described by the COBOL PICTURE.
- Zoned decimal (Z) The number of characters described by the COBOL PICTURE.
- Integer (I) 2 or 4, corresponding to decimal lengths of 1-4 or 5-9 in the COBOL PICTURE.
- Floating-point (F) 4 bytes
- Double-precision (D) 8 bytes
- Packed decimal (P) (number of PICTURE digits / 2) + 1; excluding sign (S) or implied decimal (V).

The USAGE attribute defines the display length. Allow for the maximum possible number of characters or digits including decimal points. You may include valid edit options without increasing the length size. The USAGE attribute also supports the field formats listed above.

To translate the COBOL field formats to FOCUS field formats, consult the table below. Minimum values are shown for FOCUS USAGE lengths.

COBOL USAGE TYPE	COBOL PICTURE	BYTES OF INTERNAL USAGE STORAGE	FOCUS ACTUAL FORMAT	FOCUS USAGE FORMT
DISPLAY	X(4)	4	A4	A4
DISPLAY	S99	2	Z2	P3
DISPLAY	9(5)V9	6	Z6	P8.1
COMP	S9	2	I2	I1
COMP	S9(4)	2	I2	I4
COMP	S9(5)	4	I4	I5
COMP	S9(9)	4	I4	I10
COMP-1	-	4	F4	F6
COMP-2	-	8	D8	D15

COBOL USAGE TYPE	COBOL PICTURE	BYTES OF INTERNAL USAGE STORAGE	FOCUS ACTUAL FORMAT	FOCUS USAGE FORMT
COMP-3	9	1	P1	P1
COMP-3	S9V99	2	P2	P5.2
COMP-3	9(4)V9(3)	4	P4	P8.3

For COMP-1 and COMP-2, allow for the maximum possible digits.

For COBOL DISPLAY fields with zoned decimal, FOCUS formats must be packed (P).

For COMP-1 and COMP-2, PICTURE is not permitted for internal floating-point formats (F and D).

Note:

For network record-types, the ACTUAL and USAGE formats of zoned CALC and zoned index fields must be described as alphanumeric (A).

Field formats are fully described in your FOCUS documentation.

Example

Creating GROUP Field EMP_NAME

The following example illustrates how to create the GROUP field EMP_NAME:

```
GROUP=EMP_NAME ,ALIAS=ENAME,USAGE=A25 ,ACTUAL=A25 , $
FIELD=FIRST_NAME,ALIAS=EFN ,USAGE=A10 ,ACTUAL=A10 , $
FIELD=LAST_NAME ,ALIAS=ELN ,USAGE=A15 ,ACTUAL=A15 , $
```

The GROUP field called EMP_NAME is composed of two fields, FIRST_NAME and LAST_NAME. Notice the USAGE and ACTUAL formats.

Example

Creating Dummy Fields

The following example illustrates how to create a dummy field between a 1-byte alphanumeric field and a double precision floating-point field:

```
SEGNAME=EMPSTATUS, $
FIELD=STATUS_CODE,SCODE,A1,A1, $
FIELD= ,A7,A7, $
FIELD=GROSS_SALES,GSales,D12.2,D8, $
```

The same name and alias can be used for all dummy fields; this example uses blanks.

Example **Creating a Filler Field in a Segment Describing an LRF Record**

The following example illustrates when to create a filler field in a segment describing an LRF record:

```
FILENAME=JOBMAST, SUFFIX=IDMS,$
SEGNAM=JOBEMP, SEGTYPE=S,$
  FIELDNAME=JOBID,ALIAS=JID,USAGE=A4,ACTUAL=A4,$
  FIELDNAME=TITLE,ALIAS=JOBT,USAGE=A20,ACTUAL=A20,$
  FIELDNAME=DESCRIPTN,ALIAS=CMTS.END,USAGE=A120,
    ACTUAL=A120,$
  FIELDNAME=START_DTE,ALIAS=STDT,USAGE=A6YMD,ACTUAL=A6,$
  FIELDNAME=FINISH_DTE,ALIAS=FIDT,USAGE=A6YMD,ACTUAL=A6,$
  FIELDNAME=SALARY_GRADE,ALIAS=GRADE,USAGE=P4,ACTUAL=Z2,$
  FIELDNAME=SALARY,ALIAS=SLRY,USAGE=P10.2,ACTUAL=P5,$
  FIELDNAME=,ALIAS=FILL.END,USAGE=A1,ACTUAL=A8,$
```

The LR record called JOB-EMPOSITION is defined to the IDMS subschema with seven fields: the first three fields are derived from a physical (DML) record-type called JOB; the last four fields, from the EMPOSITION physical record-type.

In this example, the filler field corresponds to the last field of the EMPOSITION record-type because LRF records must lie on a double-word boundary. The IDMS filler field is stored to make the record length a multiple of eight. IDMS filler fields on physical record-types underlying LRF views must become filler fields in the Master File. These filler fields can have any field name or be blank; the ALIAS must include the .END suffix. Since the JOB record-type is 144 bytes (total of ACTUAL formats), it does not need filler; the last field, DESCRIPTN, only requires an alias with an .END suffix.

The IDMS Database Key

The key of a network record-type corresponding to a segment can optionally be described as the last field in the segment. To specify a database key, use the following values:

Attribute	Valid Values
FIELDNAME=	Any unique name, 66 characters maximum (as long as the alias has 12 characters maximum).
ALIAS=	DBKEY
USAGE=	I10
ACTUAL=	I4

The key is used to select records from entry segments when screening on particular IDMS values in the user request. For a discussion of entry record retrieval, see Chapter 8, *CA-IDMS Record Retrieval Intervals*.

Remote Segment Descriptions

SEGTYPEs KLU and KL specify unique and non-unique segments for which the aggregate fields are remotely described in another Master File. In other words, a KLU or KL segment contains no field information; its fields are fully defined in another Master File. The CRFILE attribute specifies the source or remote Master File. At execution time, the remote field descriptions are, in effect, copied into the current Master.

KLU and KL segment types are used when several Master Files are constructed for the same IDMS subschema. They save typing and maintenance effort. Use KLU if the underlying segment is unique (U); KL if the underlying segment is non-unique (S). Remote descriptions have no logical implications regarding parent/descendant relationships nor do they impact their implementation.

Example

Specifying a Remote Segment Description

The following example illustrates how to specify a remote description:

```
SEGMENT=SALES , PARENT=DEPT , SEGTYPE=KL , CRFILE=RECORDS , $
```

To specify a remote description, you must give the KLU or KL segment the same name as the source segment in the remote Master File to ensure the proper use of field descriptions. The source Master File is specified as the CRFILE value.

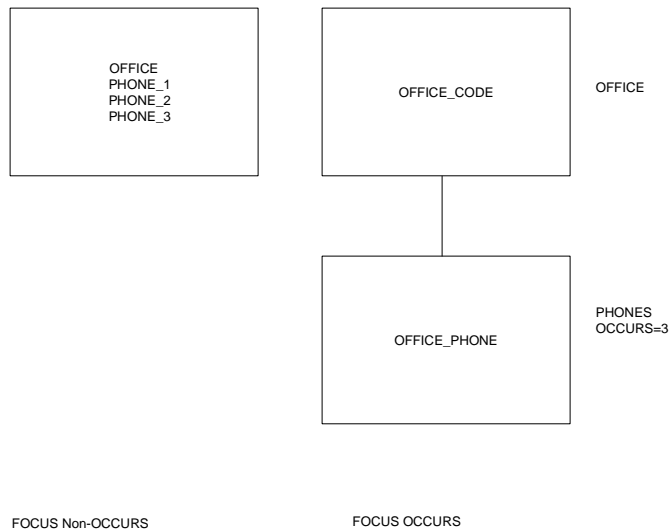
This segment declaration indicates that the field descriptions for the SALES segment are obtained from the SALES segment in the RECORDS Master File. The SEGTYPE for the SALES segment in the RECORDS Master cannot be KL or KLU, it must be U or S. Only field names and their attributes from the source file are used; original segment attributes are not. The source file does not have to be a Master File as long as it describes the named segment.

Intra-Record Structures: The OCCURS Segment

A common record structure is one in which a field or group of adjacent fields is repeated in the same record-type. In COBOL and PL/I syntax, repeating intra-record structures are defined using an OCCURS keyword. The equivalent in a Master File is an OCCURS segment.

For example, the OFFICE record-type contains the field OFFICE-PHONE that occurs three times (see Appendix B, *CA-IDMS Data Adapter Samples*). The corresponding OFFICE segment can list three separate fields with different field names, or OFFICE-PHONE can be described in a separate descendant OCCURS segment. Using the OCCURS method, each office phone is referenced by a single name (see the Master File for EMPSS01 in Appendix B, *CA-IDMS Data Adapter Samples*).

The following example depicts the FOCUS OCCURS method and the non-OCCURS method.



OCCURS segments have two attributes: ORDER and POSITION. Briefly, the POSITION attribute points to OCCURS segments when non-repeating fields exist between repeating fields (see *The POSITION Attribute* on page 4-12). The ORDER attribute creates a virtual counter field that you can use in requests (see *The ORDER Field* on page 4-13).

Describing Repeating Groups

Any fixed- or variable-length record-type described in COBOL can be mapped to a Master File hierarchy using OCCURS segments. A simple OCCURS segment is a descendant of the parent segment where the parent segment contains the non-repeating fields found in the IDMS record-type. You must specify the OCCURS segment on the descendant segment declaration that describes the repeating group. Like the COBOL OCCURS keyword, the value of the OCCURS attribute can be a numeric constant or a field name. The numeric constant indicates a fixed number of repetitions; a field name indicates a count field in the parent segment that maintains a count of the number of occurrences.

OCCURS segments also describe parallel and nested intra-record hierarchical structures. Parallel sets of repeating groups are described as multiple descendant segments of the same parent. In a nested structure, where a repeating group contains another repeating group, one OCCURS segment is the parent of another. Fixed and variable OCCURS segments can be intermixed in any order.

The restrictions for OCCURS segments are as follows:

- The count field for a variably occurring repeating group must be located physically before the repeating group in the parent of the OCCURS segment.
- A record structure that has a variable number of occurrences but no count field is not within the scope of the data adapter.
- The SEGTYPE for an OCCURS segment is specified as S or KL, indicating that it is a non-unique segment.
- OCCURS segments must be defined in the Master File in the same order as they appear in the actual record-type, unless the POSITION attribute is used (see *The POSITION Attribute* on page 4-12).
- OCCURS segments do not have a corresponding segment declaration in the Access File, because the data adapter simulates them using the parent record. OCCURS segments do not generate any additional IDMS calls.

Example

Processing OCCURS Segments

The following example illustrates how OCCURS segments are processed:

COBOL budget record-type:

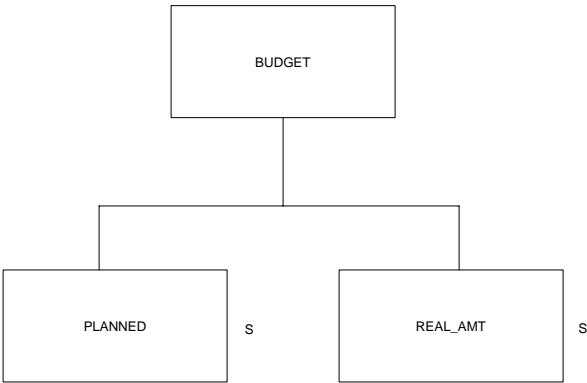
```
01  BUDGET-RCD.
    02  ACCOUNT                PIC XXX.
    02  ACTUAL-COUNT           PIC 99.
    02  PLANNED-AMT            PIC 9(9) OCCURS 12 TIMES.
    02  ACTUAL-AMT             PIC 9(9) OCCURS 12 TIMES
                                DEPENDING ON ACTUAL-COUNT.
```

OCCURS segments are indistinguishable from other segments. They are processed, if referenced, in the usual top-to-bottom left-to-right retrieval order. The equivalent Master File is:

```
SEGMENT=BUDGET, SEGTYPE=S, $
  FIELD=ACCOUNT, ALIAS=ACCT, USAGE=A3, ACTUAL=A3, $
  FIELD=ACTUAL_COUNT, ALIAS=ACTCNT, USAGE=P4, ACTUAL=Z2, $
SEGMENT=PLANNED, PARENT=BUDGET, SEGTYPE=S, OCCURS=12, $
  FIELD=PLANNED_AMT, ALIAS=PLNAMT, USAGE=P12, ACTUAL=Z9, $
SEGMENT=REAL_AMT, PARENT=BUDGET, SEGTYPE=S,
  OCCURS=ACTUAL_COUNT, $
  FIELD=ACTUAL_AMT, ALIAS=ACTAMT, USAGE=P12, ACTUAL=Z9, $
```

The mapping principle is very simple: the non-repeating field in the record-type is described in one parent segment, and the parallel COBOL OCCURS structures are specified as descendant OCCURS segments. The OCCURS attribute on the descendant segment specifies either a number (fixed occurrences) or a count field (variable occurrences). In this case, the count field ACTUAL_COUNT is located in the immediate parent segment called BUDGET and is specified in the OCCURS attribute of the REAL_AMT descendant segment.

The diagram for this Master File example follows:



If the PLANNED or REAL_AMT segments had repeating structures, they would in turn be parents of OCCURS segments defined by the same principles. The BUDGET segment could have other non-OCCURS descendants, too. The PLANNED and REAL_AMT segments might have CALC- or index-based descendants. However, set-based descendants of this record-type would be tied to the BUDGET segment, not to the PLANNED or REAL_AMT segments.

The POSITION Attribute

OCCURS segments must be defined in the same order as they appear in the actual record-type. In some cases, COBOL OCCURS structures are separated by non-repeating fields. The POSITION attribute in a Master File indicates that the repeating fields are located in the middle of non-repeating fields. The POSITION attribute is only valid for a repeating group with a fixed number of occurrences.

Example Using the POSITION Attribute

The following example illustrates how to use the POSITION attribute. Suppose the previous COBOL record-type looked like this:

```
01  BUDGET-RCD.
   02  ACCOUNT                PIC XXX.
   02  ACTUAL-COUNT            PIC 99.
   02  PLANNED-AMT             PIC 9(9) OCCURS 12 TIMES.
   02  ACTUAL-AMT              PIC 9(9) OCCURS 12 TIMES
                                DEPENDING ON ACTUAL-COUNT.
```

The POSITION attribute can only be used for a repeating group with a fixed number of occurrences. This means that the value of the OCCURS attribute of the descendant segment must be a numeric constant and not a count field.

Here, the two repeating fields PLANNED-AMT and ACTUAL-AMT are separated by the non-repeating field ACTUAL-COUNT, which clearly belongs to the BUDGET segment. You must indicate in the Master File that the first occurrence of the PLANNED segment will not immediately follow the ACCOUNT field in the BUDGET segment. (The PLANNED-AMT field is described in a separate descendant segment.) The POSITION attribute accomplishes this task by directing FOCUS to the descendant segment named PLANNED.

The corresponding Master File follows:

```
SEGMENT=BUDGET, SEGTYPE=S, $  
  FIELD=ACCOUNT, ALIAS=ACCT, USAGE=A3, ACTUAL=A3, $  
  FIELD=PLANNED_SEG1, ALIAS=PLSEG, USAGE=A108, $  
    ACTUAL=A108, $  
  FIELD=ACTUAL_COUNT, ALIAS=ACTCNT, USAGE=P4, ACTUAL=Z2, $  
SEGMENT=PLANNED, PARENT=BUDGET, SEGTYPE=S, OCCURS=12, $  
  POSITION=PLANNED_SEG1, $  
  FIELD=PLANNED_AMT, ALIAS=PLNAMT, USAGE=P12, ACTUAL=Z9, $  
SEGMENT=REAL_AMT, PARENT=BUDGET, SEGTYPE=S, $  
  OCCURS=ACTUAL_COUNT, $  
  FIELD=ACTUAL_AMT, ALIAS=ACTAMT, USAGE=P12, ACTUAL=Z9, $
```

The POSITION attribute in the PLANNED segment names a field called PLANNED_SEG1 in BUDGET, its immediate parent segment. PLANNED_SEG1 in the parent coincides with the first field of the first occurrence in PLANNED, the OCCURS segment. The REAL_AMT segment does not require a POSITION attribute, because the position of its first occurrence is correctly inferred as following the last described field in the BUDGET segment.

In this example, PLANNED_SEG1 spans all occurrences of the PLANNED segment. As an alternative, 12 individual fields named PLANNED_SEG1 through PLANNED_SEG12 could be described in the BUDGET segment. Each individual field would need the appropriate numeric format. Then reference could be made to any one of the 12 amount fields by its specific name or generically through the PLANNED_AMT field. The POSITION attribute can always be used for this purpose, even when it is not required for positioning the first position of an OCCURS segment located in the middle of the record-type.

The ORDER Field

In an OCCURS segment, the order of the fields may be significant. Using the example in *The POSITION Attribute* on page 4-12, the REAL_AMT and the PLANNED segments are most likely ordered by month. However, the record-type itself does not identify the month to which the amounts apply. ORDER is a virtual field that assigns a sequence number to each field within the repeating group.

The rules for the ORDER field are:

1. It is the last field described in an OCCURS segment.
2. The field name for it is arbitrary but its ALIAS must be ORDER (ALIAS=ORDER).

- 3. Its USAGE format type must be integer (USAGE=In) and can have edit options; its ACTUAL attribute must be I4 (ACTUAL=I4).

Since the ORDER values are 1, 2, 3, and so on, you can use the DECODE function to decode the values into months or some other meaningful value.

Example

Using the ORDER Field

The following example defines the PHONE_TYPE field by decoding the ORDER field and then uses PHONE_TYPE in a request:

```
DEFINE FILE EMPFULL
PHONE_TYPE/A8 = DECODE OCC07_01(1 'MAIN' 2 'AUX' 3 'SPEED' ELSE ' ');
END
TABLE FILE EMPFULL
PRINT OFFICE_PHONE AS 'PHONE NUMBER'
DEPT_NAME AS 'DEPARTMENT'
BY EMP_LAST_NAME AS 'LAST NAME'
BY EMP_FIRST_NAME AS 'FIRST NAME'
WHERE DEPT_ID GT '5000'
WHERE PHONE_TYPE EQ 'MAIN'
END
```

The output is:

LAST NAME	FIRST NAME	PHONE NUMBER	DEPARTMENT
-----	-----	-----	-----
ANDALE	ROY	3679191	BRAINSTORMING
ARM	HARRY	3679191	BRAINSTORMING
BREEZE	C.	3679191	BRAINSTORMING
CLOTH	TERRY	4578123	THERMOREGULATION
CLOUD	BETH	4578123	BLUE SKIES
CROW	CAROLYN	4578123	BRAINSTORMING
DONOVAN	ALAN	4578123	BLUE SKIES
FINN	PHINEAS	3697721	THERMOREGULATION
KASPAR	JOE	3679191	THERMOREGULATION
LANCHESTER	BURT	4578123	BRAINSTORMING
MAKER	RENE	4578123	BRAINSTORMING
MOON	DANIEL	4578123	BLUE SKIES
MUNYON	RICHARD	3679191	BRAINSTORMING
TIME	MARK	3679191	THERMOREGULATION
WAGNER	RICHARD	3679191	BRAINSTORMING
WILCO	ROGER	3679191	THERMOREGULATION

CHAPTER 5

CA-IDMS Data Adapter Access Files

Topics:

- Access File Overview
- Access File Syntax
- Subschema Declaration Attributes
- Segment Attributes for Network Record-Types
- Index Declarations for Network Record-Types
- Dynamically Setting the DBNAME and DICTNAME

An Access File is required to translate a report request for network record-types and LRF records into the appropriate IDMS DML retrieval commands. The Access File consists of 80-character records called declarations in comma-delimited format (attribute=value). For TSO, these records may be numbered in columns 73 through 80. See *Access File Syntax* on page 5-2 for more information.

Access File Overview

An Access File contains three kinds of declarations:

- Subschema
- Segment
- Index

Subschema declarations identify: the subschema used, the IDMS release under which the subschema was compiled, the calling mode to be used to retrieve records, and whether a trace is to be produced. Several subschema declarations can be specified in a single Access File. Each subschema declaration is followed by its segment and index declarations. See *Subschema Declaration Attributes* on page 5-3 for more information.

Access segment declarations indicate the IDMS record information and the parent/descendant relationship for each network record-type or LRF record described as a segment in a Master File. (Access segment declarations are not defined for OCCURS segments.) Segment declarations can be specified in any order after their corresponding subschema declaration. See *Segment Attributes for Network Record-Types* on page 5-5 for more information.

Index declarations provide information about each IDMS index. They may also be specified in any order following their corresponding subschema declarations, one for each indexed field described in the Master File. For further information, see *Index Declarations for Network Record-Types* on page 5-9.

This chapter contains brief examples of syntax. For more detailed examples, consult Appendix B, *CA-IDMS Data Adapter Samples*.

Access File Syntax

Each declaration consists of a list of attribute and value pairs, separated by commas. The list is free form and can span several lines; attributes can be specified in any order. Each declaration ends with a comma followed by a dollar sign (\$).

Example **Creating an Access File Containing Three Declarations**

The following example illustrates how to create an Access File with declarations consisting of attribute and value pairs:

```
SSHEMA=PAYROLL,RELEASE=14.0,INDEXAREA=PRIMARY-IX-AREA,$
SEGNAME=ACCOUNT,RECORD=PAYREC,AREA=PAY-REGION,
      CLCFLD=EMPLOYEE_ID,CLCDUP=N,$
IXSET=IXREC-SSN,IXAREA=IX-AREA1,IXFLD=PERS_SSN,
      IXDUP=N,IXORD=A,$
```

Blank lines and lines starting with a dollar sign (\$) in column 1 are treated as comments. Leading and trailing blanks around attributes and values are ignored. Values that contain commas, equal signs, dollar signs, or spaces must be enclosed in single quotation marks.

Subschema Declaration Attributes

Subschema declarations for DML and LRF subschemas contain the following attributes; certain attributes are optional as explained in the summary chart below.

Attribute	Values can be:
SSHEMA=	IDMS subschema name.
RELEASE=	14.x (where x is your release version).
MODE=	LR or DML (DML is the default).
TRACE=	YES, PARMS, or NO (NO is the default).
READY=	ALL or null or omit entirely.
DBNAME=	IDMS data source name from DBNAME table corresponding to its subschema. Used in local or CV mode. This attribute can be set dynamically; see <i>Dynamically Setting the DBNAME and DICTNAME</i> on page 5-9 for information.
DICTNAME=	Secondary dictionary load area for CV mode only. This attribute can be set dynamically; see <i>Dynamically Setting the DBNAME and DICTNAME</i> on page 5-9 for information.
NODE=	IDMS data dictionary table entry that identifies the DDS node location of an IDMS distributed database. For CV mode only.
DICTNODE=	IDMS data dictionary table entry that identifies the DDS node location of an IDMS distributed database subschema in a secondary dictionary load area. Use with CV mode only.
INDEXAREA=	Name of the integrated index area name.

If your Master File defines record-types and LRF records from one or more IDMS subschemas, the Access File should contain one or more subschema declarations. After each subschema declaration, list its segment and index declarations. You must also specify the release of the IDMS software that was used in the last compilation of the subschema. Since MODE indicates the type of IDMS access the data adapter performs, specify LR for LR records; DML is the default.

The TRACE attribute used for debugging purposes is optional. It specifies whether a basic trace of all IDMS calls or a detailed trace of all the parameters passed to IDMS will be displayed. The default is set for no trace. For large data sources, use TRACE with the IF READLIMIT command.

Another optional attribute is READY. Specify READY when an LRF record is built from two or more physical record-types located in several data source areas. The data adapter prepares or readies all the areas of the subschema.

The DBNAME attribute is optional. Its value is the IDMS data source name from the DBNAME table that locates the real subschema name and location. This attribute supports multiple data source access. It can be used in local or CV mode to translate the subschema name into the proper load modules for data access.

If the subschema is not located in the primary dictionary or in a load PDS, use the DICTNAME attribute to identify a secondary dictionary load area. Remember that ASF subschemas are located in secondary dictionary load areas by default; so, if your Access File describes an ASF record, you must specify this attribute. Access Files containing this attribute are used in IDMS CV mode only.

The NODE attribute supports IDMS Distributed Database Systems. The value for NODE is the IDMS data dictionary table entry that identifies the DDS node location of an IDMS distributed database. This attribute is required only if DDS is installed at a user site and if the subschema and data source are located in a remote site location.

The DICTNODE attribute is similar to NODE, but DICTNODE supports secondary dictionaries in DDS environments. The value of DICTNODE is the IDMS data dictionary table entry that identifies the DDS node location of an IDMS distributed database subschema. DICTNODE is required only if DDS is installed at the user site and if the subschema and data source are located in a remote site location.

Note:

When running using DDS, CV must be used because CA-IDMS does not support DDS access in local mode.

Segment Attributes for Network Record-Types

Your use of attributes in segment declarations for DML record-types depends on whether the record-type contains a CALC key, acts as a descendant segment, or contains an index.

Syntax

How to Specify Segment Attributes for Network Record-Types

The following attributes are common to all segment declarations:

`SEGNAM=segname, RECORD=record, AREA=area`

where:

segname

Is the corresponding Master File segment name of a DML record-type.

record

Is the IDMS record-type name.

area

Is the IDMS area name that contains the record-type.

Syntax

How to Specify Segment Attributes for CALC Record-Types

If your record-type is a CALC record-type (contains a CALC key) include the two attributes below.

`CLCFLD=clcfld, CLCDUP=clcdup`

where:

clcfld

Is the Master File field name of the CALC field.

clcdup

Is Y or N, depending on whether the CALC field allows duplicates.

Note:

These attributes are required for all CALC record-types, regardless of how their parent/descendant relationships are implemented.

As discussed in Chapter 3, *IDMS Overview and Mapping Concepts*, record-types are assigned parent/descendant relationships. These relationships are based on sets and CALC or index fields. Consult your Master File to determine if a segment is a descendant or parent.

Syntax

How to Specify Attributes for Set-Based Segments

You can declare attributes for set-based segments as follows.

```
ACCESS=set, SETNAME=setname, SETMBR=setmbr, GETOWN={Y|N},  
MULTMBR={Y|N}, KEYFLD=keyfld, SETORD={A|D}, SETDUP={Y|N}
```

where:

set

Is the set-based relationship. Required for descendants.

setname

Is the name of the set relating segment to its parent.

setmbr

Can be one of the following:

MA Set membership is mandatory/automatic.

MM Set membership is mandatory/manual.

OA Set membership is optional/automatic.

OM Set membership is optional/manual.

This information is used to verify set membership at execution time. To determine the appropriate value, check the set label on your Entity Relationship diagram.

GETOWN={Y|N}

Allows or inhibits GET OWNER calls, which obtain the owner records from a member record-type.

If the value is Y, the data adapter issues GET OWNER calls to retrieve the owner record in the set when SEGTYPE is U, KLU, or when an S or KL segment is inverted (the parent/descendant relationship is reversed).

If the value is N, GET OWNER calls are inhibited. Specify N only when the set has no owner pointers. When GET OWNER calls are inhibited, the owner record-type cannot be a descendant of its member. In other words, if GET OWNER calls are inhibited, SEGTYPE cannot be U or KLU and, for certain segments in inversions, record retrieval will fail.

MULTMBR={Y|N}

Y indicates that the set in which this record-type participates contains more than one member record-type.

keyfld

Is the 1- to 12-character field name or alias of the sequencing field for the set. If the key is composed of multiple fields, separate the individual field names or aliases with slashes. Required for sorted sets.

SETORD={A|D}

A indicates ascending, D indicates descending for sorted set sequence; required.

`SETDUP={Y|N}`

Is Y if duplicates are allowed, N if not. Required for sorted sets.

Syntax

How to Specify Attributes for CALC-Based Segments

You can declare attributes for CALC-based segments as follows.

`ACCESS=calc, KEYFLD=keyfld`

calc

Is the CALC-based relationship. Required for descendants.

keyfld

Is the 1- to 12-character field name or alias from the parent segment. If the key is composed of multiple fields, separate the individual field names or aliases with slashes.

Syntax

How to Specify Attributes for Index-Based Segments

You can declare attributes for index-based segments as follows.

`ACCESS=ix, KEYFLD=keyfld, SETNAME=setname`

where:

ix

Is the index-based segment. Required for descendants.

keyfld

Is the 1- to 12-character field name or alias from the parent segment. If the key is composed of multiple fields, separate the individual field names or aliases with slashes.

setname

Is the set name of the IDMS index set.

Reference

Usage Notes for Access File Attributes

The ACCESS attribute indicates the relationship that exists between record-types. The value CLC or IX specifies an embedded cross-reference based on a CALC or indexed field. The value SET indicates a physical relationship based on a set of pointers.

For a sorted set, the KEYFLD value is the FOCUS field that sequences the set. Select a FOCUS field from a parent or descendant segment as the value of KEYFLD in the descendant segment declaration.

For CALC- and index-based relationships, the KEYFLD attribute provides the search value to read CALC and index fields in descendant segments. These search values are located in parent segment fields. Specify the parent field name for the value of KEYFLD in the descendant segment declaration.

The KEYFLD attribute is especially important when the two record-types in a parent/descendant relationship are from different subschemas. The record-type that acts as the descendant segment is the entry point into the second subschema. It must have a CALC key (CLCFLD) or index set (SETNAME) with ACCESS=CLC or ACCESS=IX. The descendant segment declaration must also list the KEYFLD value from the parent segment in the first subschema.

For index-based relationships, the SETNAME value is the IDMS name of the index set. A corresponding index declaration is required (see *Index Declarations for Network Record-Types* on page 5-9 for additional information).

Syntax

How to Suppress Area Sweeps

If your record-type contains an indexed field, you may suppress area sweeps when the segment is used as a point of entry into the data source. To prevent area sweeps, specify the optional SEGFLD attribute in the segment declaration. Only those record instances connected to the specified index field are accessed.

```
SEQFIELD=seqfld
```

where:

```
seqfld
```

Is the FOCUS field name (FIELDTYPE=I) of the index.

This optional attribute requires an index declaration (see *Index Declarations for Network Record-Types* on page 5-9 for more information).

Example

Describing CALC-Based Record Types

This brief syntax example shows one subschema with two segments that are CALC record-types; the INVOICE record-type has a set-based relationship with the CUSTOMER record-type.

```
SSHEMA=SAMPSSCH,RELEASE=14,$
SEGNAM=CUSTOMER,RECORD=CUSTOMER,AREA=CUSTOMER-REGION,
  CLCFLD=CUST_NUMBER,CLCDUP=N,$
SEGNAM=INVOICE,RECORD=INVOICE,AREA=INVO-REGION,
  CLCFLD=INV_NUMBER,CLCDUP=N,ACCESS=SET,
  SETNAME=CUSTOMER-INVO,SETMBR=MA,GETOWN=Y,MULTMBR=N,$
```

The next example shows two subschemas. The INSURANCE-PLAN record-type has a CALC-based relationship with the COVERAGE record-type.

```
SSHEMA=EMPSS01,RELEASE=14,$
SEGNAM=COVERAGE,RECORD=COVERAGE,AREA=INS-DEMO-REGION,$
SSHEMA=EMPSS03,RELEASE=14,$
SEGNAM=INSURANCE,RECORD=INSURANCE-PLAN,
  AREA=INS-DEMO-REGION,CLCFLD=INS_PLAN_CODE,CLCDUP=N,
  ACCESS=CLC,KEYFLD=COV_CODE,$
```


Index Declarations for Network Record-Types

The data adapter supports indexing using the IDMS Integrated Indexes.

Syntax

How to Declare Attributes for Integrated Indexes

The following attributes apply to declarations for Integrated Indexes. This declaration is not used in LRF Access Files. See *Access File Syntax* on page 5-2 for an example of an index declaration.

```
IXSET=ixset, IXFLD=ixfld, IXDUP={Y|N}, IXORD={A|D} [, IXAREA=ixarea]
```

where:

ixset

Is the IDMS set name of the index set.

ixfld

Is the corresponding 1- to 12-character field name or alias with FIELDTYPE = I. If the key is composed of multiple fields, separate the individual field names or aliases with slashes.

```
IXDUP={Y|N}
```

Is Y if duplicate index values are allowed, N if not.

```
IXORD={A|D}
```

A indicates ascending, D indicates descending sort order of index.

ixarea

Is the IDMS area name of index. The IXAREA attribute is omitted unless the index entries reside in a different area from record-type being indexed.

Dynamically Setting the DBNAME and DICTNAME

The commands SET DBNAME and SET DICTNAME enable you to dynamically change the DBNAME and DICTNAME parameters at any time during your FOCUS session.

If you do not issue these commands, the data adapter reads the DBNAME and DICTNAME from the Access File. However, once you issue them, the DBNAME and DICTNAME values take precedence over those in your Access Files. The new values remain in effect until you either:

- Reissue the SET commands with new DBNAME and DICTNAME values.
- End your FOCUS session.
- Reinstate the Access File parameters by issuing the SET commands with the DEFAULT option.

Syntax

How to Set the DBNAME and DICTNAME

```
TSO IDMSR SET DBNAME {dbname|DEFAULT}
```

```
TSO IDMSR SET DICTNAME {dictname|DEFAULT}
```

where:

dbname

Is the IDMS database name that you want to access.

dictname

Is the IDMS dictionary name that you want to access.

DEFAULT

Causes the data adapter to read the value from the Access File.

Syntax

How to Display the Current Settings

To display the settings that are currently in effect, issue the following command:

```
TSO IDMSR SET ?
```

Example

Dynamically Changing the DBNAME and DICTNAME Values

The following example changes the DBNAME and DICTNAME parameters:

```
TSO IDMSR SET DBNAME EMPDEMO
```

```
TSO IDMSR SET DICTNAME APPLDICT
```

CHAPTER 6

Creating File Descriptions With AUTOIDMS

Topics:

- AUTOIDMS Prerequisites
- How to Execute AUTOIDMS
- The Descriptions Generated by AUTOIDMS
- How to Restart AUTOIDMS
- AUTOIDMS Sample Sessions

Two methods exist for creating Master and Access Files.

- One method is to create these files with your system editor, typing the declarations described in Chapter 4, *CA-IDMS Data Adapter Master Files*, and Chapter 5, *CA-IDMS Data Adapter Access Files*. Use an Entity Relationship diagram and a detailed report from the IDMSRPTS utility (see Appendix B, *CA-IDMS Data Adapter Samples*) as guides.
- The second method is to execute the AUTOIDMS facility. The AUTOIDMS Facility automatically creates Master and Access Files for network and LRF records. Minor editing may be required after the descriptions are generated.

This chapter describes the AUTOIDMS facility.

For instructions on how to install the AUTOIDMS facility, consult the *CA-IDMS Data Adapter Installation Guide*.

AUTOIDMS Prerequisites

AUTOIDMS automatically creates Master Files and Access Files for the CA-IDMS network and LRF records, based on information stored in the IDMS Integrated Data Dictionary and user selections.

AUTOIDMS requires disk space to write the new file descriptions (and parameter log file, if applicable) to the MVS data sets specified on the initial input screen or to the disk accessed as file mode A in VM. In addition, sufficient disk space must be available for temporary work files created. The amount of temporary space depends on the size of the subschema being described.

For MVS execution, the Master Files IDMSIDD and AUTOIDMS must be members in the sequence of data sets assigned to the DDNAME MASTER. The IDMSIDD Access File must be a member in the sequence of data sets assigned to the DDNAME FOCIDMS. The AUTOIDMS FOCEXEC must be a member in the sequence of data sets assigned to the DDNAME FOCEXEC. These members should be located in your site's production FOCUS libraries as a result of the data adapter installation process.

For VM/CMS execution, the files AUTOIDMS FOCEXEC, AUTOIDMS MASTER, IDMSIDD MASTER, and IDMSIDD FOCIDMS should all be on a mini-disk available to the user. This is usually the FOCUS production disk, as a result of the data adapter installation process.

Before you begin, you should decide which record-types are appropriate for your Master File. Consider which record should be the root segment and which record/set relationships will be included as descendants. A review of the Entity Relationship diagram for the subschema will assist in these decisions. Since you can create several Master Files in the same AUTOIDMS session, with only one pass through the IDMS dictionary, you may want to prepare several views of the subschema before proceeding.

How to Execute AUTOIDMS

To start AUTOIDMS, enter the FOCUS environment and issue the following command at the FOCUS prompt:

```
EX AUTOIDMS
```

AUTOIDMS allocates or FILEDEFs some temporary data sets/files.

Next, the Main Menu opens. (At this point, if you saved data sets/files in a previous session, you are prompted with the Restart Screen. See *How to Restart AUTOIDMS* on page 6-2 for more information.

The AUTOIDMS Main Menu

The following is an example of the AUTOIDMS main menu in MVS/TSO. User entries are in lower case.

```

Main Menu                                Master File Generation Facility for IDMS/R
                                         Information Builders, Inc.

IDMS Subschema Name to be Used =====> empss01
Master Filename =====> testidms
DBName =====> empdemo DictName =====> appldict
Secondary Dictionary=>          ( A)

Description will be a member of:
  Master Target PDS => PMSJSC.MASTER.DATA
  Access Target PDS => PMSJSC.FOCIDMS.DATA
  Replace Existing Description? ===> N      (Y/N)
  Exclude Comments from MFD?   ===> N      (Y/N)
  Exclude Description from MFD? ===> N      (Y/N)

Field Information
  Use Record ID Suffix in Name? ===> Y      (Y/N)
  Start with Element Name Position => 1      (1-32)
                                for a Total Length of => 32 (1-32)
  Describe Numeric Displays as   => A      (N-Numeric,A-Alpha)
  Parm File => PMSJSC.FOCIDMS.DATA

PF1=Help PF3=Exit PF4=Log PF5=TED MFD PF6=TED AFD PF9=Picture PF10=List

```

The following is an example of the AUTOIDMS main menu in VM/CMS. User entries are in lower case:

```
Main Menu      Master File Generation Facility for IDMS/R
                Information Builders, Inc.

IDMS Subschema Name to be Used =====> empss01
Master Filename =====> testidms

Replace Existing Description?  ==>  N      (Y/N)
Exclude Comments from MFD?    ==>  N      (Y/N)
Exclude Description from MFD?  ==>  N      (Y/N)

Field Information
Use Record ID Suffix in Name? ==>  Y      (Y/N)
Start with Element Name Position =>  1      (1-32)
                                for a Total Length of => 32      (1-32)
Describe Numeric Displays as   =>  N      (N-Numeric,A-Alpha)

Parm File =>  IDMS$PRM FOCIDMS A

PF1=Help  PF3=Exit  PF4=Log  PF5=TED MFD  PF6=TED AFD  PF9=Picture PF10=List
```

With the exception of the main menu, AUTOIDMS screens are identical in MVS/TSO and VM/CMS.

On the main menu, provide information about the subschema to be described, a Master File name to use in report requests, and target data sets for the descriptions generated (MVS only). The field information parameters allow you to customize the field name attributes produced by AUTOIDMS.

Specify the following information on the AUTOIDMS Main Menu:

IDMS Subschema Name	Enter the 1- to 8-character subschema name that you wish to describe. The subschema must already exist in the Data Dictionary.
Master Filename	Enter the 1- to 8-character name that you will use to refer to the data in all requests. (In MVS, this name must be a valid member name. In CMS, this name must be a valid file name.)
DBName	IDMS database name. Required when DictName is entered
DictName	IDMS data dictionary.
Secondary Dictionary	Dictionary suffix for the IDMSIDD. A indicates ascending, D indicates descending sort order.

Master Target PDS	(MVS only) Enter the fully qualified data set name of the Master File PDS where the Master File will be stored. Do not use quotation marks in the data set name.
Access Target PDS	(MVS only) Enter the fully qualified data set name of the Access File PDS where the Access File will be stored. Do not use quotation marks in the data set name.
Replace Existing Description? (Y/N)	Enter N if you do not want to overwrite an existing Master File/Access File. Enter Y if you wish to replace a Master File/Access File that already exists on disk.
Exclude Comments from Mfd? (Y/N)	Enter Y to exclude commented entries from the Master File. AUTOIDMS will comment non-supported dictionary entries such as re-defined elements, groups within groups, and 88 level elements. Enter N to retain the commented elements.
Exclude Description from Mfd? (Y/N)	Enter Y to exclude the \$DESC=entry from the Master File. Enter N to include the description (excluding comments will also exclude the description).
Use Record Id Suffix in Name? (Y/N)	Enter Y to use the record element synonym name as the FIELDNAME (using SYN-NAM-083 from the dictionary). Enter N to use the primary element name as the FIELDNAME (using DR-NAM-042). The synonym name contains the schema record ID as a suffix (for example, STATUS-0415); the primary name does not (for example, STATUS). Excluding the record ID suffix may cause duplicate names when more than one record is described.
Start with Element Name Position (1-32)	Enter the starting position of the element name to be used as the first character in the FIELDNAME. By increasing the starting position, you may strip off common prefixes from the element name.
for a Total Length of (1-32)	<p>Enter the number of characters of the element name to be used in the FIELDNAME. Entering 32, for example, will use the entire element name (when the starting position is 1). When the starting position is greater than 1, the FIELDNAME length will be the smaller of the total length provided on the main menu or the remaining length of the element name.</p> <p>Using a length less than 32 may cause truncation of the field name and duplicate field names in the resulting Master File.</p>

Describe Numeric
Displays as (N-
Numeric,A-Alpha)

Enter N to have elements with an IDMS DISPLAY of PIC 9(*n*) described as numeric values with USAGE=*In* in the Master File. Enter A to have elements with an IDMS DISPLAY of PIC 9(*n*) described as alphanumeric with USAGE=*An* in the Master File.

If you intend to use COBOL numeric displays for summing or mathematical equations, use the Numeric option. If you intend to use these elements for display only, use the Alphanumeric option. If these elements are part of a GROUP field, you may want to use the Alphanumeric option so that the GROUP field will be described for easy access in the Master File.

The following functions are available from the Main Menu using the PF keys:

- Help (PF1)

Press PF1 to display an extensive online help facility; much of the text of this document is available using this function.

- Exit (PF3)

Press PF3 to exit AUTOIDMS.

- Logging Default Menu Parameters (PF4)

Press PF4 to save your customized default values for the Main Menu to disk. In MVS, these will be saved in member IDMS\$PRM in the parm data set shown on the menu. This data set will be: the PDS preallocated to ddname IDMS\$PRM, 'userid.FOCIDMS.DATA', or the first data set allocated to ddname FOCIDMS (see *Search Order for the AUTOIDMS Parameter Log File (MVS only)* on page 6-16 for further details). Parameter logging assumes that the user has write access to the target data set. An attempt to log parameters to a data set without write access will result in a security abend. In CMS, these will be saved to file name IDMS\$PRM FOCIDMS A.

The following information will be logged:

- IDMS Subschema Name
- Dictionary
- DBNAME
- Secondary Dictionary
- Master File Partitioned Data Set Name (MVS only)
- Access File Partitioned Data Set Name (MVS only)
- Replace existing description
- Exclude Comments from Master File
- Exclude Description from Master File
- Use Record ID Suffix
- Starting Field name Position
- Total Field name Length

- Describe Numeric Displays

All validations must pass before the default values are logged.

- TED MFD (PF5)/TED AFD (PF6)

Press PF5 or PF6 to edit (using TED) the Master or Access File, respectively, entered in Master File name on the main menu. In MVS, this will select the member in the data set named in either MASTER TARGET PDS or ACCESS TARGET PDS. In CMS, this will select *mastername* MASTER A or *mastername* FOCIDSM A. These files did not have to be created with AUTOIDMS to edit them.

- PICTURE OF MFD (PF9)

Press PF9 to generate a diagram of the structure of the file entered in Master File name. After the picture is displayed, type in any character and press Enter. In MVS, the Master name must be a member of a data set allocated to ddname MASTER to generate the picture (the MASTER TARGET PDS is not used).

- Subschema List (PF10)

Leave the subschema name blank and press PF10 to display a list of all subschemas described in the IDMS Data Dictionary. A subset of the list can be generated by entering a partial name with wild card characters.

An underscore (_) may be used as a mask to select any character in the position used. The percent sign (%) may be used as a mask to select any sequence of zero or more characters.

All validations must pass in order to use these PFkey options.

Enter the appropriate values on the initial screen, and press Enter. AUTOIDMS will inform you that it is accessing the IDMS Integrated Data Dictionary by issuing the following message:

```
**=====**
**  AUTOIDMS is retrieving information from IDD  **
**  Please wait...                             **
**=====**
```

This message is only displayed during the first retrieval in an AUTOIDMS session, or when the subschema criteria have changed since the previous retrieval (within the session). AUTOIDMS will not access the dictionary a second time for the same immediate subschema in a single session.

When subschema information retrieval has been completed, the Root Record Selection Screen will be displayed.

The AUTOIDMS Root Record Selection Screen

The Root Record Selection is used to choose the first record of the selected subschema to be described. It will be the root segment in the generated Master File.

The following is the Root Record Selection screen for the EMPSS01 subschema. User entries are in lower case:

```

Subschema :EMPSS01      Master File Generation Facility for IDMS/R
Master      :TEST      ==Root Record Selection==
Dictionary:
Place an 'X' next to the record to be the root of the Master

X Record Name                                Index Type  Calc?  ID
- - - - -                                - - - - -  - - - -
  COVERAGE                                NONE        N      0400
  DENTAL-CLAIM                            NONE        N      0405
  DEPARTMENT                              NONE        Y      0410
X EMPLOYEE                                INTEGRATED  Y      0415
  EMPOSITION                              NONE        N      0420
  EXPERTISE                               NONE        N      0425
  HOSPITAL-CLAIM                          NONE        N      0430
  INSURANCE-PLAN                          NONE        Y      0435
  JOB                                     INTEGRATED  Y      0440
  NON-HOSP-CLAIM                          NONE        N      0445
  OFFICE                                  NONE        Y      0450
  SKILL                                   INTEGRATED  Y      0455

PF1=Help      PF3=End      PF7=Up
PF8=Down

```

Type X at the record desired and press Enter to select the root.

It is recommended that the root record-type have a set relationship with a CALC or index field. Record-types with logical (CALC or index-based) relationships require special attention (see *Special Considerations for AUTOIDMS* on page 6-17 for further details).

Once the root is selected, the Record/Set Selection screen will be displayed for set based records that participate in a set. For records that are not part of a set, or for LRF records, the description will be generated and the Main Menu will be redisplayed.

The Root Record Selection Screen displays the following information:

Subschema	Subschema name provided on the Main Menu.
Master	Master File name provided on the Main Menu.
X	Select the record to be chosen as the root by placing an X in this column.
Record Name	Record name from the IDMS Data Dictionary. LRF records will have 'LR=Y' appended to the record name.

<code>Index Type</code>	Indicates the type of index on the record. Values are NONE (no index), INTEGRATED, or blank (LRF records).
<code>Calc?</code>	Indicates whether a CALC field exists. Values are N (no), Y (yes), or blank (LRF records).
<code>ID</code>	The IDMS record ID. For LRF records the value is LRF.

The following functions are available from the Root Record Selection screen using the PF keys:

- **Help (PF1)**
Press PF1 to display an extensive online help facility; much of the text of this document is available using this function.
- **Exit (PF3)**
Press PF3 to exit the file description generator and return to the main menu.
- **Scroll Backward (PF7)/Scroll Forward (PF8)**
Press PF7 to scroll the record list backward; press PF8 to scroll the record list forward.

When the root record has been selected, the Record/Set Selection Screen will be displayed.

The AUTOIDMS Record/Set Selection Screen

The Record/Set Selection Screen is used to choose the descendants of the record shown on the top of the screen.

The following shows the Record/Set Selection Screen for record EMPLOYEE, which was chosen as the root. User entries are in lower case.

```

Subschema :EMPSS01      Master File Generation Facility for IDMS/R      Segments
Dictionary:
Place an 'X' next to each record/set to be a descendant of SEGNAME: S0415_01
Record: EMPLOYEE

Parent:                               Set:

X Record Name                        Set Name                        Owner/ Own No.
- - - - -                          - - - - -                      Member Ptr Occ
X DEPARTMENT                        DEPT-EMPLOYEE                        O   Y
  COVERAGE                        EMP-COVERAGE                        M   Y
  EMPOSITION                        EMP-EMPOSITION                        M   Y
X EXPERTISE                        EMP-EXPERTISE                        M   Y
  STRUCTURE                        MANAGES                             M   Y
X OFFICE                          OFFICE-EMPLOYEE                      O   Y   1
  STRUCTURE                        REPORTS-TO                          M   Y

PF1=Help  PF2=Restart  PF3=End  PF4=None  PF5=Picture  PF7=Up  PF8=Down

```

Type X next to the records you want to describe as descendants and press Enter. Press PF4 if none of the records/sets are to be described as descendants of the record listed at the top of the screen.

In the sample above, the EMPLOYEE record-type is the current record-type; its segment name is S0415_01. Relationship-types for EMPLOYEE vary. For example, DEPARTMENT acts as the owner of EMPLOYEE in the DEPT-EMPLOYEE set, while EXPERTISE acts as the member in EMP-EXPERTISE set.

The record relationships will be described in top down, left to right order. Initially, this screen will display the records and sets that the root record participates in. Then, it will show the records/sets of the first selection of the previous screen. This continues until the selected record does not participate in any other sets or the user selects no descendant records (by pressing PF4). Then the records/sets of the second selection of the previous screen are shown for selection. Once all possible descendant selections have been exhausted, the description will be generated and the Main Menu will be redisplayed.

The Record/Set Selection Screen displays the following information:

Subschema	Subschema name provided on the Main Menu.
Master	Master File name provided on the Main Menu.
Segments Left	The number of records (including OCCURS) segments that can still be described, up to a maximum of 64 segments.

Segname	The segment name for the record displayed (generated by AUTOIDMS).
Record	The name of the record already described on a previous screen for which the records/sets are being displayed.
Parent	The parent of the record shown. This is displayed to illustrate the top down order of the selection process.
Set	Set name that relates the displayed record and its parent (blank for the root segment).
X	Select the records to be chosen as descendants by placing an X in this column.
Record Name	Record name related (using Set Name) to the record shown above.
Set Name	Set name that relates the record on this line with the record shown above. The set will not appear in this list.
Owner/Member	Indicates whether the record on this line is the owner (O) or member (M) of this set.
Own Ptr	Owner pointer indicator. Y indicates that the member record has a pointer to the owner of the set. Generally, do not select descendant sets where the Owner/Member flag is O and the owner pointer is N unless the member record chain is short. (GET OWNER calls are issued to retrieve these relationships and can adversely effect data retrieval efficiency.)
No. Occ	Number of OCCURS (repeating groups) described in this record.

The following functions are available from the Record/Set Selection Screen using the PF keys:

- Help (PF1)
Press PF1 to display an extensive online help facility; much of the text of this document is available using this function.
- Restart (PF2)
Press PF2 to cancel all previous selections and restart the Master File generation process at the Root Record Selection Screen.
- Exit (PF3)
Press PF3 to exit the file description generator and return to the main menu.
- No Record/Set Selections (PF4)
Press PF4 if you do not wish to describe any of the displayed record/sets as descendants of the current record.

- Creating a Picture of the Description (PF5)

Press PF5 at any time to generate a diagram of the structure of the data source being described.

Note that the description will be created on disk (and will remain there even if the program is ended with PF3).

After the picture is displayed, type any character and press the Enter key. In MVS, the target Master PDS must be allocated to ddname MASTER to generate the picture. If a member exists with the Master name selected in a data set concatenated in front of the target data set, the picture will be generated from that member.

- Scroll Backward (PF7)/Scroll Forward (PF8)

Press PF7 to scroll the record list backward; press PF8 to scroll the record list forward.

The Descriptions Generated by AUTOIDMS

AUTOIDMS creates complete Master Files and Access Files for the subschema and records selected by the user. The following describes the attributes and values assigned to them by AUTOIDMS.

File and Segment Attributes Generated by AUTOIDMS

The file and segment attributes are:

<code>FILE=</code>	Master File name specified on the Main Menu.
<code>SEGNAME=</code>	For records described in the Master File, the segment name is composed of the letter S, the four digit record id number, an underscore, and a two digit sequence number to ensure uniqueness (for example, S0415_01). For repeating groups within a record, the segment name is composed of the letters OCC, the two-digit sequence number of the parent segment, an underscore, and its own two-digit sequence number (for example, OCC04_01).
<code>SEGTYPE=</code>	For member segments, the value is S; for owner segments, the segtype is U. The segtype for repeating groups is S.
<code>PARENT=</code>	For dependant segments, this is the value of the SEGNAME attribute of the parent record.
<code>OCCURS=</code>	For repeating groups only, this will contain either the fixed number of occurrences of the repeating group or the name of the field that contains the number of variable occurrences of the group.

POSITION= For repeating groups only, this entry identifies the position of the repeating values in the parent record. If the repeating values are described as a GROUP, this value will be the ALIAS of the group field. If the repeating group consists of only one field, this value will be the letter S, the two digit sequence number of the parent, the string _OCC, and the two digit sequence number of the occurs segment (for example, S04_OCC01). This artificial name is used as the placeholder in the parent record so that the field name can be used in the occurs segment.

Field Attributes Generated by AUTOIDMS

The field attributes are:

FIELD=	This attribute identifies simple IDMS elements as fields.
GROUP=	This attribute identifies IDMS group elements that are composed of other contiguous fields. The GROUP attribute is only used for groups at the highest level in the record description.
\$GROUP=	This attribute identifies IDMS group elements that are themselves components of another group. Since these are embedded groups, they cannot be referenced. However, you can reference this field by uncommenting the entry (by removing the leading dollar sign) and commenting the parent group (by adding a leading dollar sign).
\$ RDF=	This attribute identifies redefined IDMS elements. You can reference either the original fields or the redefined fields. You can reference the redefined field by changing \$ RDF= to FIELD= and commenting the original field (by adding a leading dollar sign).
\$ 88 =	This attribute identifies conditional IDMS elements (COBOL level 88 fields). These entries are provided in the Master File for documentation purposes only. The value of the conditional field is contained in the USAGE attribute. If the conditional field is a range, the low end of the range is in the USAGE attribute; the high end of the range is in the ACTUAL attribute.
\$ BIT=	This attribute identifies BIT fields. The first BIT field in a set is described with a FIELD attribute and an alphanumeric format. The length is the number of bytes occupied by all BIT fields in the set. Subsequent BIT fields in the set are described as comments with the \$ BIT= attribute. To access the individual BIT fields, you can create virtual fields (within the Master File or a FOCEXEC) using the BITSON subroutine, using the first field in the set as input.

<code>\$DESC=</code>	This attribute lists the full IDMS element name. If the field name is truncated as a result of user selections, this will also include the string <code>*TRUNC*</code> .
<code>\$DUPLICATE</code>	This attribute, found at the end of the Master File, identifies duplicate fields described in the Master File. It lists the full IDMS element name, the number of duplicate occurrences, and the segments in which the duplicate fields are located.
Field names	Field names are derived from the IDMS dictionary and may be truncated based on user selections on the main menu. Hyphens in the IDMS name are converted to underscores. Certain field names are generated by AUTOIDMS. A field that is generated to ensure full-word boundaries at the end of a record in an LRF will be blank. A calculated position field will have the format <code>Smm_OCCnn</code> (see the description of the <code>POSITION</code> attribute in Chapter 4, <i>CA-IDMS Data Adapter Master Files</i>). A field that describes the internal IDMS database key (whose alias is <code>DBKEY</code>) will consist of the <code>SEGNAME</code> and the string <code>_KEY</code> . A field indicating the sequence number for a repeating group (whose alias is <code>ORDER</code>) will be the same as its <code>SEGNAME</code> .
Aliases	Alias names are calculated by AUTOIDMS. For most fields, this value will be the letter <code>F</code> , the four digit record ID, an underscore, and a unique sequence number (for example, <code>F0415_1</code>). If a record is described more than once in a Master File, the alias will be suffixed with an alphabetic character. The alias will therefore be unique within a Master File and will be consistent from Master File to Master File for the same subschema. If truncation of field names causes duplicates, the unique alias can be used to reference the field instead. The alias for the last field of each record in an LRF will consist of the letter <code>F</code> , the two-character segment sequence number, and the string <code>.END</code> . The alias for a calculated position field will be blank. The alias <code>DBKEY</code> is used to identify the internal IDMS database key value and appears at the end of each IDMS record described. The alias <code>ORDER</code> is used to identify the sequence number of a repeating group and appears at the end of each segment for repeating groups.

Changes to the Descriptions Generated by AUTOIDMS

In general, you will not need to change the descriptions generated by AUTOIDMS. You may need to edit the generated Master File and/or its corresponding Access File if:

- Duplicate field names exist. Truncation or ring structures in which record-types are used more than once are the likely causes.
- A record-type is described that contains a discontinuous key comprised of more than eight fields. In this case, shorten the aliases of the root fields in the Master File and Access File, and append the aliases of the additional fields in the Access File.
- You prefer to use different field names than those in the IDMS dictionary.
- You prefer to use an embedded GROUP field instead of the group at the highest level.
- You prefer to use redefined fields in place of the parent fields.
- You wish to display the values of BIT fields.
- You wish to add display options to fields.
- You wish to increase the display length of numeric fields.

Creating a Record of Master Files Generated by AUTOIDMS

AUTOIDMS maintains a temporary list of the Master Files generated during any one session. This list is refreshed at the beginning of each session and is normally erased at the end of the session.

Syntax

How to Create Records of Master Files Generated

You can retain a list of Master Files generated by executing AUTOIDMS with the following syntax:

```
EX AUTOIDMS MFDLIST=Y
```

In MVS, the list will reside in a temporary data set allocated to ddname AUTOIDML, with a disposition of MOD and record length of 114.

In CMS, the list will reside in the file AUTOIDML FOCTEMP A with a record length of 26. It is the user's responsibility to free or erase this file when no longer required. The file layout is:

Length	Columns	Description
8	1 – 8	Master File name
8	9 –16	Number of duplicate field names
10	17 – 26	Blank
44	27 – 70	Master Target PDS name (MVS only)
44	77 – 114	Access Target PDS name (MVS only)

Search Order for the AUTOIDMS Parameter Log File (MVS only)

The AUTOIDMS parameter log file in MVS will be located in one of three ways, in the following order:

1. DDNAME IDMS\$PRM

- If this ddname is allocated to a PDS prior to execution of AUTOIDMS, it will be used. The member IDMS\$PRM will be used or created for the user.
- If this ddname is allocated to a sequential data set, it will be freed, a message generated, and parameter logging will be disabled.
- AUTOIDMS does not free this ddname upon exiting, assuming that it may be used again, even if it was allocated by AUTOIDMS with one of the following two methods.

2. Data set name *userid.FOCIDMS.DATA*

- This is the default name provided in the code as &DSNP0. If that default has been changed during the data adapter installation, then that data set name will be used.
- This data set must be a PDS. If not, a message is displayed and parameter logging is disabled.

3. DDNAME FOCIDMS

- The first data set allocated to ddname FOCIDMS will be used as the parameter file if the first two methods fail.
- If ddname FOCIDMS is not allocated, a message is displayed and AUTOIDMS exits. AUTOIDMS cannot be executed if there is no FOCIDMS ddname for the IDMSIDD files.

Method number two assumes standard Information Builders naming conventions. Method number three assumes that a user's data set is allocated first in the concatenation of data sets to ddname FOCIDMS. The first method allows the user to identify the profile data set prior to execution of AUTOIDMS. This option is recommended for sites that have non-standard data set naming conventions.

How to Restart AUTOIDMS

If, in a previous session, you stopped AUTOIDMS and saved the data sets/files (with PF4), you can continue to process them or save them for another session.

After you execute AUTOIDMS, you are prompted with:

```

MASTER FILE GENERATION FACILITY FOR IDMS
** INFORMATION BUILDERS, INCORPORATED **
DURING A PREVIOUS SESSION THE GENERATION OF MASTER EMPFULL1

FOR SUBSCHEMA EMPSS01 WAS PREMATURELY HALTED.

ENTER 'R' TO RESTART PROCESSING THE ABOVE MASTER
'S' TO SAVE THE FILES AND GENERATE ANOTHER MEMBER
OR 'D' TO DELETE THE FILES =====> R
PRESS PF3 KEY TO TERMINATE PROCEDURE

```

Specify one of the following and press Enter:

- R** Continues processing.
- S** Saves the pre-existing data sets/files and allocates/FILEDEFs more data sets/files.
- D** Deletes the pre-existing data sets/files and allocates/FILEDEFs more data sets/files.

Reference

Special Considerations for AUTOIDMS

In AUTOIDMS processing, only network record-types with set-based relationships can be generated as descendant segments. To produce file descriptions that include record-types with CALC- or index-based relationships (ACCESS=CALC or ACCESS=INDEX), you must:

1. Generate two pairs of file descriptions: one pair for set-based record-types and one pair for CALC-based record-types.
2. Copy one pair into the other.
3. Edit the pair of resulting file descriptions. For example, add the KEYFLD parameter to the Access File to create the embedded cross-reference.

An alternate method is to use the JOIN command to dynamically cross-reference the two Master Files at execution time (see Chapter 7, *CA-IDMS Data Adapter Reporting Techniques*, for more information).

AUTOIDMS Sample Sessions

These two sample sessions will create the Master and Access Files used in the examples found throughout the manual.

Creating the EMPFULL Master and Access Files

Start the AUTOIDMS procedure by issuing the following command:

```
EX AUTOIDMS
```

The main menu opens.

Entering Information on the AUTOIDMS Main Menu

Enter the following information:

- The IDMS subschema name, EMPSS01.
- A name for the Master File. We will choose the name EMPFULL.
- The IDMS DBNAME, EMPDEMO.
- The IDMS DICTNAME, APPLDICT.

Also, to make the Master File shorter and more readable, change the following default entries:

- Exclude Comments from MFD. Change to Y.
- Exclude Description from MFD? Change to Y
- Use Record ID Suffix in Name? Change to N. This choice will generate more duplicate names in the resulting Master File. Duplicate names and the segments to which they apply are listed as comments at the bottom of the generated Master File. You can edit the Master File to make the names unique, or you can qualify the fieldname with its corresponding segment name in request.

The following shows the main menu with the choices specified:

```

Main Menu                                Master File Generation Facility for IDMS/R
                                         Information Builders, Inc.

IDMS Subschema Name to be Used =====> empss01
Master Filename =====> empfull
DBName =====> empdemo   DictName =====> appldict
Secondary Dictionary=>      ( A)

Description will be a member of:
Master Target PDS => PMSSAE.MASTER.DATA
Access Target PDS => PMSSAE.FOCIDMS.DATA
Replace Existing Description? =====> N      (Y/N)
Exclude Comments from MFD? =====> y      (Y/N)
Exclude Description from MFD? =====> y      (Y/N)

Field Information
Use Record ID Suffix in Name? =====> n      (Y/N)
Start with Element Name Position => 1      (1-32)
                                for a Total Length of => 32      (1-32)
Describe Numeric Displays as => A      (N-Numeric,A-Alpha)
Parm File => PMSSAE.FOCIDMS.DATA

PF1=Help PF3=Exit PF4=Log PF5=TED MFD PF6=TED AFD PF9=Picture PF10=List

```

Press Enter. The following message displays:

```

**=====**
**      AUTOIDMS is retrieving information from IDD      **
**      Please wait...                                  **
**=====**

```

Selecting DEPARTMENT as the Root Segment With AUTOIDMS

When AUTOIDMS finishes retrieving information from the IDD, the Root Record Selection screen opens.

Select DEPARTMENT to be the root segment in the generated Master File:

Subschema :EMPSS01 Master File Generation Facility for IDMS/R

Master :EMPFULL ==Root Record Selection==

Dictionary:

Place an 'X' next to the record to be the root of the Master

X Record Name	Index Type	Calc?	ID
- - - - -	- - - - -	- - - - -	- - - - -
COVERAGE	NONE	N	0400
DENTAL-CLAIM	NONE	N	0405
x DEPARTMENT	NONE	Y	0410
EMPLOYEE	INTEGRATED	Y	0415
EMPOSITION	NONE	N	0420
EXPERTISE	NONE	N	0425
HOSPITAL-CLAIM	NONE	N	0430
INSURANCE-PLAN	NONE	Y	0435
JOB	INTEGRATED	Y	0440
NON-HOSP-CLAIM	NONE	N	0445
OFFICE	NONE	Y	0450
SKILL	INTEGRATED	Y	0455

PF1=Help PF3=End PF7=Up PF8=Down

Selecting Descendant Segments for EMPFULL With AUTOIDMS

When you press Enter, the Record/Set Selection screen opens. Select EMPLOYEE to be the descendant of DEPARTMENT:

Subschema :EMPSS01 Master File Generation Facility for IDMS/R Segments

Dictionary:

Place an 'X' next to each record/set to be a descendant of SEGNAME: S0410_01

Record: DEPARTMENT

Parent: Set:

X Record Name	Set Name	Owner/ Own No.
- - - - -	- - - - -	Member Ptr Occ
- - - - -	- - - - -	- - - - -
x EMPLOYEE	DEPT-EMPLOYEE	M Y

PF1=Help PF2=Restart PF3=End PF4=None PF5=Picture PF7=Up PF8=Down

When you press Enter, the Record/Set Selection screen opens so you can select descendants for EMPLOYEE. Select COVERAGE, EMPOSITION, EXPERTISE, OFFICE, and STRUCTURE (from the MANAGES set):

```

Subschema :EMPSS01      Master File Generation Facility for IDMS/R      Segments
Dictionary:
Place an 'X' next to each record/set to be a descendant of SEGNAME: S0415_02
Record: EMPLOYEE
Parent: DEPARTMENT                               Set: DEPT-EMPLOYEE

X Record Name                               Set Name                               Owner/ Own No.
- - - - -                               - - - - -                               Member Ptr Occ
- - - - -                               - - - - -                               - - - - -
x COVERAGE                               EMP-COVERAGE                               M   Y
x EMPOSITION                             EMP-EMPOSITION                             M   Y
x EXPERTISE                             EMP-EXPERTISE                             M   Y
x STRUCTURE                             MANAGES                                    M   Y
x OFFICE                                OFFICE-EMPLOYEE                           O   Y   1
      STRUCTURE                           REPORTS-TO                                M   Y

PF1=Help  PF2=Restart  PF3=End  PF4=None  PF5=Picture      PF7=Up  PF8=Down

```

AUTOIDMS next displays the Record /Set Selection screen for COVERAGE. Select DENTAL-CLAIM, HOSPITAL-CLAIM, and NON-HOSP-CLAIM as descendants:

```

Subschema :EMPSS01      Master File Generation Facility for IDMS/R      Segments
Dictionary:
Place an 'X' next to each record/set to be a descendant of SEGNAME: S0400_03
Record: COVERAGE
Parent: EMPLOYEE                               Set: EMP-COVERAGE

X Record Name                               Set Name                               Owner/ Own No.
- - - - -                               - - - - -                               Member Ptr Occ
- - - - -                               - - - - -                               - - - - -
x DENTAL-CLAIM                           COVERAGE-CLAIMS                           M   N   1
x HOSPITAL-CLAIM                         COVERAGE-CLAIMS                           M   N   1
x NON-HOSP-CLAIM                         COVERAGE-CLAIMS                           M   N   2

PF1=Help  PF2=Restart  PF3=End  PF4=None  PF5=Picture      PF7=Up  PF8=Down

```

Next Select JOB as the descendant of EMPOSITION:

Subschema :EMPSS01 Master File Generation Facility for IDMS/R Segments

Dictionary:

Place an 'X' next to each record/set to be a descendant of SEGNAME: S0420_04

Record: EMPOSITION

Parent: EMPLOYEE Set: EMP-EMPOSITION

X Record Name	Set Name	Owner/ Own No. Member Ptr Occ
- - - - -	- - - - -	- - - - -
x JOB	JOB-EMPOSITION	O Y 3

PF1=Help PF2=Restart PF3=End PF4=None PF5=Picture PF7=Up PF8=Down

Select skill as the descendant of EXPERTISE

Subschema :EMPSS01 Master File Generation Facility for IDMS/R Segments

Dictionary:

Place an 'X' next to each record/set to be a descendant of SEGNAME: S0425_05

Record: EXPERTISE

Parent: EMPLOYEE Set: EMP-EXPERTISE

X Record Name	Set Name	Owner/ Own No. Member Ptr Occ
- - - - -	- - - - -	- - - - -
x SKILL	SKILL-EXPERTISE	O Y

PF1=Help PF2=Restart PF3=End PF4=None PF5=Picture PF7=Up PF8=Down

Select EMPLOYEE as the descendant of STRUCTURE. This employee segment will be for the subordinates of the ones selected previously:

```

Subschema :EMPSS01      Master File Generation Facility for IDMS/R      Segments
Dictionary:
Place an 'X' next to each record/set to be a descendant of SEGNAME: S0460_06
Record: STRUCTURE
Parent: EMPLOYEE                               Set: MANAGES

X Record Name                               Set Name                               Owner/ Own No.
- - - - -                               - - - - -                               Member Ptr Occ
x EMPLOYEE                                REPORTS-TO                                O   Y

PF1=Help PF2=Restart PF3=End PF4=None PF5=Picture PF7=Up PF8=Down

```

Next, the Record/Set Selection screen for EMPLOYEE displays again. There are no descendant segments, so press PF4:

```

Subschema :EMPSS01      Master File Generation Facility for IDMS/R      Segments
Dictionary:
Place an 'X' next to each record/set to be a descendant of SEGNAME: S0415_13
Record: EMPLOYEE
Parent: STRUCTURE                               Set: REPORTS-TO

X Record Name                               Set Name                               Owner/ Own No.
- - - - -                               - - - - -                               Member Ptr Occ
DEPARTMENT                                DEPT-EMPLOYEE                                O   Y
COVERAGE                                EMP-COVERAGE                                M   Y
EMPOSITION                                EMP-EMPOSITION                                M   Y
EXPERTISE                                EMP-EXPERTISE                                M   Y
STRUCTURE                                MANAGES                                    M   Y
OFFICE                                  OFFICE-EMPLOYEE                                O   Y   1

PF1=Help PF2=Restart PF3=End PF4=None PF5=Picture PF7=Up PF8=Down

```

Generating the EMPFULL Master and Access Files

When you press PF4, AUTOIDMS creates the Master and Access Files. The main menu displays with the message DESCRIPTION CREATED at the bottom:

```

Main Menu                                Master File Generation Facility for IDMS/R
                                         Information Builders, Inc.

IDMS Subschema Name to be Used =====> EMPSS01
Master Filename =====> EMPFULL
DBName =====> EMPDEMO   DictName =====> APPLDICT
Secondary Dictionary=>      ( A )

Description will be a member of:
Master Target PDS => PMSSAE.MASTER.DATA
Access Target PDS => PMSSAE.FOCIDMS.DATA
Replace Existing Description? ===> Y      (Y/N)
Exclude Comments from MFD?    ===> Y      (Y/N)
Exclude Description from MFD? ===> Y      (Y/N)

Field Information
Use Record ID Suffix in Name? ===> N      (Y/N)
Start with Element Name Position => 1      (1-32)
      for a Total Length of => 32      (1-32)
Describe Numeric Displays as   => A      (N-Numeric,A-Alpha)
Parm File => PMSSAE.FOCIDMS.DATA
DESCRIPTION CREATED - 126 DUPLICATE FIELDNAMES FOUND
PF1=Help PF3=Exit PF4=Log PF5=TED MFD PF6=TED AFD PF9=Picture PF10=List
    
```

Note that 126 duplicate field names were created. If these names are in separate segments, you can qualify them with the segment name in requests. Alternatively, you can edit the Master File to create unique field names. All of the duplicate names and their corresponding segments are listed as comments at the bottom of the Master File.

Editing the Generated EMPFULL Master File

To view or edit the generated Master File using TED from within AUTOIDMS, press PF5:

```

PMSSAE.MASTER.DATA(EMPFULL)          SIZE=446   LINE=0

00000 * * * TOP OF FILE * * *
00001 $$$ CREATED BY AUTOIDMS ON 02/13/01 AT 14.32.16 BY PMSSAE
00002 $$$ SSHEMA=EMPSS01
00003 FILE=EMPFULL,SUFFIX=IDMSR,$
00004
00005 SEGNAME=S0410_01,SEGTYPE=S,$
00006     FIELD=DEPT_ID                ,ALIAS=F0410_1    ,A4    ,A4    ,$
00007     FIELD=DEPT_NAME              ,ALIAS=F0410_2    ,A45   ,A45   ,$
00008     FIELD=DEPT_HEAD_ID           ,ALIAS=F0410_3    ,A4    ,A4    ,$
00009     FIELD=FIL 0003               ,ALIAS=F0410_4    ,A3    ,A3    ,$
00010     FIELD=S0410_01_KEY           ,ALIAS=DBKEY      ,I10   ,I4    ,$
00011
00012 SEGNAME=S0415_02,SEGTYPE=S,PARENT=S0410_01,$
00013     FIELD=EMP_ID                 ,ALIAS=F0415_1    ,A4    ,A4    ,$
00014     GROUP=EMP_NAME               ,ALIAS=F0415_2    ,A25   ,A25   ,$
00015     FIELD=EMP_FIRST_NAME         ,ALIAS=F0415_3    ,A10   ,A10   ,$
00016     FIELD=EMP_LAST_NAME          ,ALIAS=F0415_4    ,A15   ,A15   ,$
00017     GROUP=EMP_ADDRESS             ,ALIAS=F0415_5    ,A46   ,A46   ,$
00018     FIELD=EMP_STREET              ,ALIAS=F0415_6    ,A20   ,A20   ,$
00019     FIELD=EMP_CITY               ,ALIAS=F0415_7    ,A15   ,A15   ,$

```

Press PF3 to return to the AUTOIDMS main menu. Appendix B, *CA-IDMS Data Adapter Samples*, shows the complete Master File.

Note:

The Master File you edit does not have to be the one generated. You can enter any name in the Master Filename field on the main menu and press PF5 to edit it.

Editing the Generated EMPFULL Access File

To view or edit the generated Access File using TED from within AUTOIDMS, press PF6:

```
PMSSAE.FOCIDMS.DATA(EMPFULL)          SIZE=95      LINE=0

00000 * * * TOP OF FILE * * *
00001   $$$ CREATED BY AUTOIDMS ON 02/13/01 AT 14.32.16 BY PMSSAE
00002   $$$ FILE=EMPFULL,SUFFIX=IDMSR,$
00003   SSHEMA=EMPSS01,RELEASE=12.0,MODE=DML,
00004   DBNAME=EMPDEMO ,DICTNAME=APPLDICT , $
00005
00006   SEGNAME=S0410_01,RECORD=DEPARTMENT,
00007   AREA=ORG-DEMO-REGION,
00008   CLCFLD=F0410_1,
00009   CLCDUP=N,$
00010
00011   SEGNAME=S0415_02,RECORD=EMPLOYEE,
00012   AREA=EMP-DEMO-REGION,
00013   ACCESS=SET,SETNAME=DEPT-EMPLOYEE,
00014   SETMBR=OA,GETOWN=Y,MULTMBR=N,
00015   CLCFLD=F0415_1,
00016   CLCDUP=N,
00017   KEYFLD=F0415_4/F0415_3,
00018   SETORD=A,SETDUP=Y,$
00019   IXSET=EMP-NAME-NDX,IXDUP=Y,IXORD=A,
```

The Access File contains the attribute RELEASE=12.0. If you are accessing a different release of IDMS, you can change this attribute.

Press PF3 to return to the AUTOIDMS main menu. Appendix B, *CA-IDMS Data Adapter Samples*, shows the complete Access File.

Note:

The Access File you edit does not have to be the one generated. You can enter any name in the Master Filename field on the main menu and press PF6 to edit it.

Viewing the EMPFULL Structure Diagram

To view a diagram of the structure, press PF9:

```
NUMBER OF ERRORS=      0
NUMBER OF SEGMENTS=  21  ( REAL=    21  VIRTUAL=    0 )
NUMBER OF FIELDS=   267  INDEXES=    0  FILES=    1
TOTAL LENGTH OF ALL FIELDS= 3597
```

SECTION 01

STRUCTURE OF IDMSR

FILE EMPFULL ON 02/13/01 AT 11.58.57

```

      S0410_01
01    S1
*****
*DEPT_ID_0410**
*DEPT_NAME_0>**
*DEPT_HEAD_I>**
*FILLER      **
*            **
*****
      I
      I
      I
      I S0415_02
02    I S1
*****
*EMP_ID_0415 **
*EMP_NAME_04>**
*EMP_FIRST_N>**
*EMP_LAST_NA>**
*            **
*****
      I
      +-----+
      I      I
      I S0450_07  I S0400_03
03    I U      05    I S1
*****          *****
*OFFICE_CODE>*  *SELECTION_D>**
*OFFICE_ADDR>*  *SELECTION_Y>**
*OFFICE_STRE>*  *SELECTION_M>**
*OFFICE_CITY>*  *SELECTION_D>**
*            *  *            **
*****          *****
      I          I
      I          I
      I          I
      I      I S0405_08  I S0430_09  I S0445_10
04    I S1      06    I S1      08    I S1      10    I S1
*****          *****          *****          *****
*OFFICE_PHON>**  *CLAIM_DATE_>**  *CLAIM_DATE_>**  *CLAIM_DATE_>**
*OCC07_01      **  *CLAIM_YEAR_>**  *CLAIM_YEAR_>**  *CLAIM_YEAR_>**
*            **  *CLAIM_MONTH>**  *CLAIM_MONTH>**  *CLAIM_MONTH>**
*            **  *CLAIM_DAY_0>**  *CLAIM_DAY_0>**  *CLAIM_DAY_0>**
*            **  *            **  *            **  *            **
*****          *****          *****          *****
      I          I          I
      I          I          I
      I          I          I
      I      I OCC08_01  I OCC09_01  I OCC10_01
      07    I S1      09    I S1      11    I S1
*****          *****          *****
*TOOTH_NUMBE>**  *DIAGNOSIS_0>**  *DIAGNOSIS_0>**
*SERVICE_DAT>**  *OCC09_01      **  *OCC10_01      **
*SERVICE_YEA>**  *            **  *            **
*SERVICE_MON>**  *            **  *            **
*            **  *            **  *            **
*****          *****          *****
      *****          *****          *****

```

SECTION 02

```
-----+-----+-----+
          I          I          I
          I S0420_04  I S0425_05  I S0460_06
13      I S1      18      I S1      20      I S1
*****          *****          *****
*START_DATE_>**  *SKILL_LEVEL>**  *STRUCTURE_C>**
*START_YEAR_>**  *EXPERTISE_D>**  *STRUCTURE_D>**
*START_MONTH>**  *EXPERTISE_Y>**  *STRUCTURE_Y>**
*START_DAY_0>**  *EXPERTISE_M>**  *STRUCTURE_M>**
*          **      *          **      *          **
*****          *****          *****

          I          I          I
          I          I          I
          I S0440_11  I S0455_12  I S0415_13
14      I U      19      I U      21      I U
*****          *****          *****
*JOB_ID_0440 *    *SKILL_ID_04>*    *EMP_ID_0415 *
*TITLE_0440 *    *SKILL_NAME_>*    *EMP_NAME_04>*
*DESCRIPTION>*    *SKILL_DESCR>*    *EMP_FIRST_N>*
*S11_OCC01 *    *S0455_12_KEY*    *EMP_LAST_NA>*
*          *      *          *      *          *
*****          *****          *****

          I
          I
-----+-----+-----+
          I          I          I          I
          I OCC10_02  I OCC11_01  I OCC11_02  I OCC11_03
12      I S1      15      I S1      16      I S1      17      I S1
*****          *****          *****          *****
*SERVICE_DAT>**  *DESCRIPTION>**  *REQUIREMENT>**  *SALARY_GRAD>**
*SERVICE_YEA>**  *OCC11_01 **      *OCC11_02 **      *OCC11_03 **
*SERVICE_MON>**  *          **      *          **      *          **
*SERVICE_DAY>**  *          **      *          **      *          **
*          **      *          **      *          **      *          **
*****          *****          *****          *****
*****          *****          *****          *****
```

TYPE ANY CHARACTER AND PRESS ENTER TO CONTINUE >

As indicated, type any character and press Enter to return to the AUTOIDMS main menu. Then press PF3 to exit AUTOIDMS.

Creating the EMPFILE Master and Access Files

The EMPFILE Master and Access Files are used in the sample traces shown in Appendix C, *Tracing CA-IDMS Data Adapter Processing*.

Start the AUTOIDMS procedure by issuing the following command:

`EX AUTOIDMS`

The main menu opens.

Entering Information on the AUTOIDMS Main Menu

Enter the following information:

- The IDMS subschema name, EMPSS01.
- A name for the Master File. We will choose the name EMPFILE.
- The IDMS DBNAME, EMPDEMO.
- The IDMS DICTNAME, APPLDICT.

Also, to make the Master File shorter and more readable, change the following default entries:

- Exclude Comments from MFD. Change to Y.
- Exclude Description from MFD? Change to Y
- Use Record ID Suffix in Name? Change to N. This choice will generate more duplicate names in the resulting Master File. Duplicate names and the segments to which they apply are listed as comments at the bottom of the generated Master File. You can edit the Master File to make the names unique, or you can qualify the fieldname with its corresponding segment name in request.

The following shows the main menu with the choices specified:

```
Main Menu                                Master File Generation Facility for IDMS/R
                                         Information Builders, Inc.

IDMS Subschema Name to be Used =====> empss01
Master Filename =====> empfile
DBName =====> empdemo   DictName =====> appldict
Secondary Dictionary=>      ( A)

Description will be a member of:
Master Target PDS => PMSSAE.MASTER.DATA
Access Target PDS => PMSSAE.FOCIDMS.DATA
Replace Existing Description? ===> N      (Y/N)
Exclude Comments from MFD?   ===> y      (Y/N)
Exclude Description from MFD? ===> y      (Y/N)

Field Information
Use Record ID Suffix in Name? ===> n      (Y/N)
Start with Element Name Position => 1      (1-32)
                               for a Total Length of => 32 (1-32)
Describe Numeric Displays as  => A      (N-Numeric,A-Alpha)
Parm File => PMSSAE.FOCIDMS.DATA

PF1=Help PF3=Exit PF4=Log PF5=TED MFD PF6=TED AFD PF9=Picture PF10=List
```

Press Enter. The following message displays:

```
**=====**
**  AUTOIDMS is retrieving information from IDD  **
**  Please wait...                               **
**=====**
```


Selecting EMPFILE Segments With AUTOIDMS

When AUTOIDMS finishes retrieving information from the IDD, the Root Record Selection screen displays. Select EMPLOYEE as the root and press Enter:

```

Subschema :EMPSS01      Master File Generation Facility for IDMS/R
Master      :EMPFILE          ==Root Record Selection==
Dictionary:
Place an 'X' next to the record to be the root of the Master

X Record Name                Index Type  Calc? ID
- - - - -
  COVERAGE                   NONE        N   0400
  DENTAL-CLAIM               NONE        N   0405
  DEPARTMENT                 NONE        Y   0410
x EMPLOYEE                   INTEGRATED   Y   0415
  EMPOSITION                 NONE        N   0420
  EXPERTISE                  NONE        N   0425
  HOSPITAL-CLAIM            NONE        N   0430
  INSURANCE-PLAN           NONE        Y   0435
  JOB                       INTEGRATED   Y   0440
  NON-HOSP-CLAIM            NONE        N   0445
  OFFICE                    NONE        Y   0450
  SKILL                     INTEGRATED   Y   0455

PF1=Help                PF3=End                                PF7=Up  PF8=Down
  
```

Next select DEPARTMENT as the descendant of EMPLOYEE and press Enter:

```

Subschema :EMPSS01      Master File Generation Facility for IDMS/R      Segments
Dictionary:
Place an 'X' next to each record/set to be a descendant of SEGNAME: S0415_01
Record: EMPLOYEE
Parent:                                Set:

X Record Name                Set Name                Owner/ Own No.
- - - - -                    - - - - -                Member Ptr Occ
x DEPARTMENT                 DEPT-EMPLOYEE                O   Y
  COVERAGE                  EMP-COVERAGE                 M   Y
  EMPOSITION                 EMP-EMPOSITION               M   Y
  EXPERTISE                  EMP-EXPERTISE                M   Y
  STRUCTURE                  MANAGES                      M   Y
  OFFICE                     OFFICE-EMPLOYEE              O   Y   1
  STRUCTURE                  REPORTS-TO                   M   Y

PF1=Help  PF2=Restart  PF3=End  PF4=None  PF5=Picture      PF7=Up  PF8=Down
  
```

```

Subschema :EMPSS01      Master File Generation Facility for IDMS/R      Segments
Dictionary:
Place an 'X' next to each record/set to be a descendant of SEGNAME: S0415_01
Record: EMPLOYEE
Parent:                  Set:

X Record Name              Set Name              Owner/ Own No.
- - - - -                - - - - -                Member Ptr Occ
x DEPARTMENT              DEPT-EMPLOYEE              O   Y
COVERAGE                 EMP-COVERAGE              M   Y
EMPOSITION                EMP-EMPOSITION            M   Y
EXPERTISE                 EMP-EXPERTISE             M   Y
STRUCTURE                 MANAGES                   M   Y
OFFICE                    OFFICE-EMPLOYEE           O   Y   1
STRUCTURE                 REPORTS-TO                M   Y

PF1=Help  PF2=Restart  PF3=End  PF4=None  PF5=Picture      PF7=Up  PF8=Down

```

The description is now generated:

```

Main Menu                                Master File Generation Facility for IDMS/R
                                         Information Builders, Inc.

IDMS Subschema Name to be Used ==>>  EMPSS01
Master Filename ==>>>>>>>>>>>>>>>  EMPFILE
DBName ==>> EMPDEMO      DictName ==>>>  APPLDICT
Secondary Dictionary=>              ( A )

Description will be a member of:
  Master Target PDS =>  PMSSAE.MASTER.DATA
  Access Target PDS =>  PMSSAE.FOCIDMS.DATA
  Replace Existing Description? ==>>  Y      (Y/N)
  Exclude Comments from MFD? ==>>  Y      (Y/N)
  Exclude Description from MFD? ==>>  Y      (Y/N)

Field Information
  Use Record ID Suffix in Name? ==>>  N      (Y/N)
  Start with Element Name Position =>  1      (1-32)
                                     for a Total Length of =>  32      (1-32)
  Describe Numeric Displays as      =>  A      (N-Numeric,A-Alpha)
  Parm File =>  PMSSAE.FOCIDMS.DATA

DESCRIPTION CREATED
PF1=Help  PF3=Exit  PF4=Log  PF5=TED MFD  PF6=TED AFD  PF9=Picture PF10=List

```

Press PF3 to exit AUTOIDMS. For a complete listing of the generated Master and Access Files, see Appendix B, *CA-IDMS Data Adapter Samples*.

CHAPTER 7

CA-IDMS Data Adapter Reporting Techniques

Topics:

- The Retrieval Subtree
- Retrieval Sequence
- Effects of Screening Conditions on Retrieval
- Retrieving Short Paths
- File Inversion
- The JOIN Command

This chapter explains how your Master File is used to select segments and retrieve data from them. It also discusses:

- How screening conditions (IF criteria) improve I/O efficiency (see *Effects of Screening Conditions on Retrieval* on page 7-5).
- Unique and non-unique segments with short paths (see *Retrieving Short Paths* on page 7-8).
- How you can change retrieval logic by specifying a new root segment (see *File Inversion* on page 7-12).
- How you can join structures together for reporting purposes (see *The JOIN Command* on page 7-14).

The examples in this chapter reference the EMPFULL Master and Access Files. Chapter 6, *Creating File Descriptions With AUTOIDMS*, contains a sample session that demonstrates how to create the EMPFULL Master and Access Files. For a complete listing of these file descriptions, see Appendix B, *CA-IDMS Data Adapter Samples*.

Note:

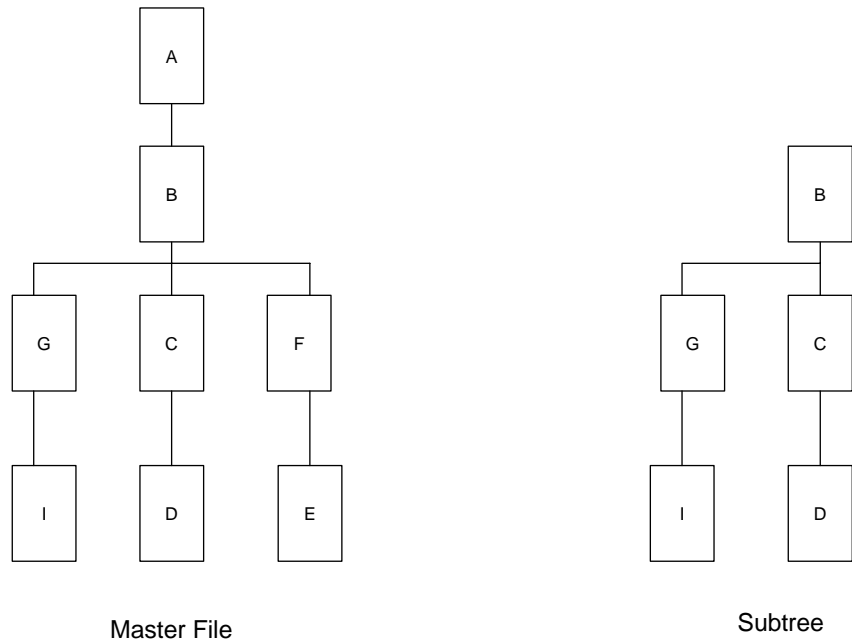
The term *record* refers to a record occurrence; it should not be confused with the IDMS term record-type.

Topics pertaining to unique segments discuss considerations/implications that are true regardless of whether the unique segment is an owner of its parent or related to it through a CALC field, index, or LRF field.

The Retrieval Subtree

To retrieve records for a request, a smaller subtree structure is constructed from the structure defined by your Master File. The subtree consists of segments that contain fields named explicitly in the request and those named implicitly by virtual fields or calculated values. The subtree also includes any segments needed to connect these segments.

For example, if a TABLE request needs fields from segments D, G, and I, the subtree shown in the following diagram is constructed:



Segments C and B do not contain fields needed for the request, but they are included in the subtree to connect segment I with segments D and G. Since the segments are descendants of segment B, the entry or root segment of the subtree is segment B. The Master File root segment A is not included, because its fields are not referenced; the records corresponding to segment B can be obtained independently of A. However, if segment B were an OCCURS segment, the IDMS calls would be issued for segment A to obtain B's information.

Retrieval Sequence

To determine the order of record retrieval, examine a diagram of the subtree. Use the CHECK FILE command to display the diagram:

```
CHECK FILE filename PICTURE
```

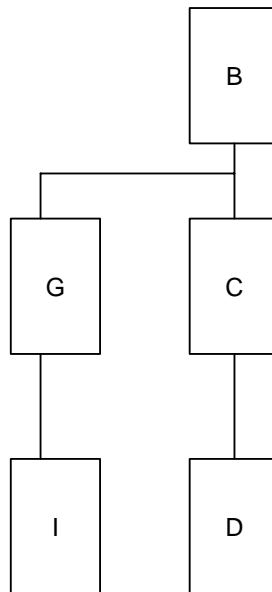
where:

filename

Is the name of the Master File.

The order of retrieval is from top to bottom, left to right. Only the first four fields of each segment are displayed.

For example, in the following figure, the first record in B is retrieved, then the first record in G, and all I records for the first G record. When all the G records and their descendant I records are retrieved, the first record in C is retrieved. Then all the D records for the first C are retrieved. When all the C records and their descendant D records are retrieved, the next record is B is retrieved and the process repeats itself.



Retrieval Sequence With Unique Segments

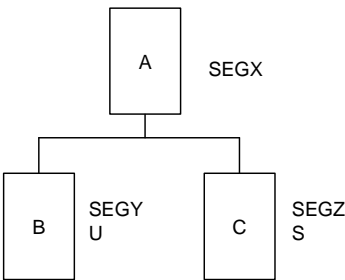
The retrieval sequence for subtrees containing unique segments is still top to bottom, left to right, but the unique segments are treated as extensions of their parents. Records in a unique segment correspond one-to-one with the records in a parent; records in a non-unique segment have a one-to-many correspondence. In cases where the parent segment has unique and non-unique descendants, the unique descendants are always retrieved first regardless of the left-to-right order.

To display a retrieval view of any subtree, use the CHECK FILE command with the RETRIEVE option:

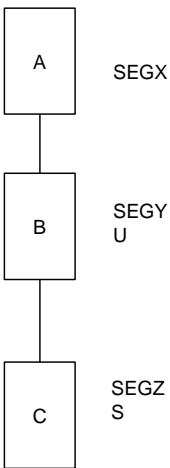
```
CHECK FILE filename PICTURE RETRIEVE
```

A retrieval view shows which sort and IF criteria are valid. For sort phrases (BY or ACROSS), the segment containing the sort field must lie on the same path as the segments with all requested fields. That is, the segment with the BY or ACROSS field must be an ancestor or descendant of the segments containing the required fields.

For example, panel 1 in the following figure shows two descendant segments. The command PRINT B BY C is invalid, because the segments containing fields B and C do not lie on the same path. However, if the segment containing B is a unique segment, the two segments do lie on the same path. Panel 2 shows the retrieval view, and the command PRINT B BY C is valid.



Panel 1



Panel 2

The retrieval sequence for unique segments may also affect the results of COUNT or SUM commands. If a segment is the parent of a unique descendant, there is a one-to-one relationship. A COUNT command like COUNT A AND B returns identical results for each field, because the same record A is counted several times for each record B.

If the parent/descendant relationship is reversed with a non-unique parent, the result for field A is a greater number than the result for field B.

Effects of Screening Conditions on Retrieval

If a record in a segment fails a record selection (IF) test, the corresponding records in descendant segments are not retrieved.

To increase I/O efficiency, place a record selection test at a higher level in the file structure. This restricts the number of records that have to be tested.

Example

Using Selection Criteria on Parent Segments

Suppose this request is entered against the structure EMPSS01 (the corresponding Master File is EMPFULL).

```
TABLE FILE EMPFULL
COUNT EMP_ID AS 'NUMBER OF EMPLOYEES'
BY OFFICE_CODE AS 'OFFICE CODE'
IF DEPT_NAME EQ 'PERSONNEL'
END
```

The output is:

```
OFFICE CODE  NUMBER OF EMPLOYEES
-----
002                                4
```

Every time a record in segment DEPT has a value in the DEPT_NAME field not equal to PERSONNEL, the corresponding records in descendant segments EMPLOYEE and OFFICE are ignored, and the next record in segment DEPT is retrieved. In addition, when IF criteria on a lower segment fail, the row is removed from the internal matrix.

Example **Using an IF Test on a Descendant Segment**

Assume a subtree has four segments:

- DEPT contains department IDs and information.
- EMPLOYEE contains employee names and IDs.
- EMPOSIT contains the positions that the employee has held.
- JOB contains a list of jobs offered by the company.

The following example illustrates how to list all employees who are programmer/analysts:

```
TABLE FILE EMPFULL
PRINT TITLE AS 'TITLE'
BY DEPT_NAME AS 'DEPARTMENT'
BY EMP_LAST_NAME AS 'LAST NAME' IN 25
BY EMP_FIRST_NAME AS 'FIRST NAME'
IF TITLE IS 'PROGRAMMER/ANALYST'
END
```

The output is:

DEPARTMENT	LAST NAME	FIRST NAME	TITLE
-----	-----	-----	-----
INTERNAL SOFTWARE	DOUGH	JANE	PROGRAMMER/ANALYST
	GALLWAY	JAMES	PROGRAMMER/ANALYST
	GRANGER	PERCY	PROGRAMMER/ANALYST
	HEAROWITZ	VLADIMIR	PROGRAMMER/ANALYST
	JENSEN	JULIE	PROGRAMMER/ANALYST
	O'HEARN	KATHERINE	PROGRAMMER/ANALYST

For this request with only one IF test, FOCUS retrieves each DEPT record, each EMPLOYEE record for a given DEPT, each EMPOSIT record for a given EMPLOYEE record, and the JOB record connected to each EMPOSIT. After retrieval, FOCUS determines whether to print the record from the value of the TITLE field.

Example

Using Multiple IF Tests to Reduce I/O

The following example produces the same output as the example in *Using an IF Test on a Descendant Segment* on page 7-6. In this company, only the Internal Software department has programmer/analysts working for it. Therefore, adding an additional IF test on the parent segment (DEPT) reduces the number of descendant segments retrieved:

```
TABLE FILE EMPFULL
PRINT TITLE AS 'TITLE'
BY DEPT_NAME AS 'DEPARTMENT'
BY EMP_LAST_NAME AS 'LAST NAME' IN 25
BY EMP_FIRST_NAME AS 'FIRST NAME'
IF DEPT_NAME IS 'INTERNAL SOFTWARE'
IF TITLE IS 'PROGRAMMER/ANALYST'
END
```

Now records are only retrieved and tested when the DEPT_NAME field equals the value INTERNAL SOFTWARE.

Restricting the Number of Records Retrieved

Another method used to increase efficiency is to restrict the number of records retrieved. You can always limit the number of records retrieved from the IDMS data source, and/or the number of records that FOCUS accepts, with READLIMIT and RECORDLIMIT tests. Add one or both of these IF criteria to your request.

Syntax

How to Place a Limit on Records Retrieved

```
IF READLIMIT IS number
IF RECORDLIMIT IS number
```

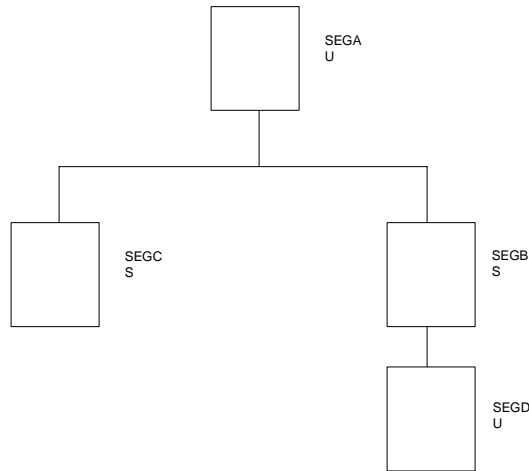
Tip:

These criteria can be assigned to your password as part of file security through the FOCUS DBA facility.

A third method that increases efficiency is file inversion. See *File Inversion* on page 7-12 for more information.

Screening Conditions With Unique Segments

If a record in a unique segment fails a selection test, its parent is rejected and the next record of the parent segment is retrieved. For example in the following figure, if a record in non-unique segment C fails an IF test, the next record in segment C is retrieved. Only if all C records for a given A fail the test is the A record rejected. When a record in unique segment D fails a test, the parent B record is rejected and the next record in segment B is retrieved. When a record in the entry A segment fails a test, FOCUS retrieves the next A record, even if the entry segment is defined as unique:



Retrieving Short Paths

When a record in a parent segment is retrieved, the corresponding records in the descendant segment are retrieved. If descendant records do not exist, the processing of the parent record depends on whether the descendant segment is unique or non-unique.

Short Paths in Unique Descendants

For a unique descendant segment with a missing record, a temporary record is created to replace the missing record. The temporary record contains fields with default values: blanks for alphanumeric fields and zeros for numeric fields.

For example, an EMPLOYEE segment with the field EMP_NAME has a unique descendant OFFICE segment with the field OFFICE_CITY. The field OFFICE_CITY indicates the location of an employee's office. Gary Smith does not work out of an office location so, he has no OFFICE record. In this situation, all reports that refer to OFFICE_CITY display blank spaces for the entry GARY SMITH.

Short Paths in Non-Unique Descendants

For a non-unique descendant segment with a missing record, the results depend on how the ALL parameter is set:

`SET ALL = {ON|OFF}`

where:

ON

The parent record is processed provided that there are no screening conditions on fields in the descendant segment. Missing data is usually indicated on the report by the default NODATA character (.). For information about overriding the NODATA default, see your FOCUS documentation.

OFF

The parent instance is rejected and the next parent instance is retrieve. OFF is the default value.

Note:

SET ALL = PASS is *not* supported by the data adapter.

Example

Using SET ALL = OFF

The EMPLOYEE segment has a non-unique descendant segment that contains dental claim records. With SET ALL = OFF, a request that prints employee names and dentist names omits employees that have no dental claims:

```
SET ALL = OFF
TABLE FILE EMPFULL
PRINT EMP_NAME DENTIST_NAME
IF DEPT_NAME EQ 'EXECUTIVE ADMINISTRATION'
               OR 'COMPUTER OPERATIONS'
END
```

The output is:

EMP_NAME		DENTIST_NAME
-----		-----
HERBERT	CRANE	DR PEPPER
HENRIETTA	HENDON	SAL SARDONICUS

Example **Using SET ALL = ON**

With SET ALL = ON, a request that prints dentist names and employee names prints everyone's name, even those without dental claims. The report output displays the nodata symbol for the field from the missing descendant segment:

```
SET ALL = ON
TABLE FILE EMPFULL
PRINT EMP_NAME DENTIST_NAME
IF DEPT_NAME EQ 'EXECUTIVE ADMINISTRATION'
                OR 'COMPUTER OPERATIONS'
END
```

The output is:

EMP_NAME		DENTIST_NAME	
-----		-----	
HERBERT	CRANE	DR	PEPPER
JANE	FERNDALE	.	
GEORGE	FONRAD	.	
ROBIN	GARDNER	.	
DOUGLAS	KAHALLY	.	
TERENCE	KLWELLEN	.	
SANDY	KRAAMER	.	
HERBERT	LIPSICH	.	
NANCY	TURNER	.	
HENRIETTA	HENDON	SAL	SARDONICUS
THEMIS	PAPAZEUS	.	
JOHN	RUPEE	.	
ROBBY	WILDER	.	

Example **Using SET ALL = ON With Screening Criteria**

When a request contains an IF test on the descendant (dental) segment, parent (employee) records are omitted if the descendant segment fails the IF test. Only Herbert Crane has a dental claim for tooth number 99:

```
SET ALL = ON
TABLE FILE EMPFULL
PRINT EMP_NAME DENTIST_NAME
IF DEPT_NAME EQ 'EXECUTIVE ADMINISTRATION'
                OR 'COMPUTER OPERATIONS'
IF TOOTH_NUMBER EQ 99
END
```

The output is:

EMP_NAME		DENTIST_NAME	
-----		-----	
HERBERT	CRANE	DR	PEPPER

Selective ALL Prefix

If the value for SET ALL is OFF, you can still apply the effect of the ON setting to specific segments. To do this, add the ALL prefix to one of the *parent* segment fields in your request. The ALL prefix causes records in that segment to be processed even if they have missing descendants. Like the SET ALL parameter, the processing depends on whether IF criteria exist for the descendant record fields.

Note:

The IF criteria provision affects only immediate descendants. In cases where descendants are missing their descendant record occurrences, the parent record occurrence is rejected.

For example, the following request displays people who have no dental claim records; however, it would not display people who have dental claim records and no corresponding dental service records.

```
SET ALL = OFF
TABLE FILE EMPFULL
PRINT ALL.EMP_NAME
DENTIST_NAME SERVICE_DATE
  IF DEPT_NAME EQ 'EXECUTIVE ADMINISTRATION'
      OR 'COMPUTER OPERATIONS'
END
```

The output is:

EMP_NAME		DENTIST_NAME		SERVICE_DATE
-----		-----		-----
HERBERT	CRANE	DR	PEPPER	19800916
HERBERT	CRANE	DR	PEPPER	19800916
JANE	FERNDAL	.		.
GEORGE	FONRAD	.		.
ROBIN	GARDNER	.		.
DOUGLAS	KAHALLY	.		.
TERENCE	KLWELLEN	.		.
SANDY	KRAAMER	.		.
HERBERT	LIPSICH	.		.
NANCY	TERNER	.		.
HENRIETTA	HENDON	SAL	SARDONICUS	19770502
THEMIS	PAPAZEUS	.		.
JOHN	RUPEE	.		.
ROBBY	WILDER	.		.

The ALL prefix is only effective when the SET ALL value is OFF.

File Inversion

When you create a Master File, you create a default representation of a hierarchy. Sometimes, though, you may not want to follow the default route to retrieve records. Two situations are:

1. Your IF criteria screen a segment at the bottom of a subtree.
2. You are processing a multi-path report with IF criteria or sort phrases that are not on a common path.

When these situations occur, you can specify a new entry segment (root) at execution time for a specific request. This process is called file inversion because the parent/descendant relationships along the path linking the original root and the new root are reversed; other parent/descendant relationships remain unchanged.

Note:

File inversions only change the file views; they do not affect the data.

Syntax

How to Invert a File

```
TABLE FILE filename.field
```

where:

```
field
```

May be any field in the new root segment.

For example, to invert the EMPFULL file so that the office segment is the new root, specify the field OFFICE_CODE:

```
TABLE FILE EMPFULL.OFFICE_CODE
```

You can also display a diagram of the inverted file with the CHECK FILE command. (Include the RETRIEVE option for a subtree diagram.):

```
CHECK FILE filename.fieldname PICTURE [ RETRIEVE]
```

You cannot invert a Master File if:

- The path linking the old and new roots passes through segments that have a CALC- or index-based relationship.
- The GETOWN parameter in the ACCESS file for a set-based relationship is set to N.
- It is an LRF Master File.

File inversion is a simple solution to two common problems:

- Denied access because the segment is on the wrong sort path.
- Denied access because the field named in an IF test is not on the root path.

Example**Using File Inversion to Solve Sort Path Problems**

In addition to solving the sort path problem, file inversion can improve I/O efficiency which, in turn, minimizes production costs.

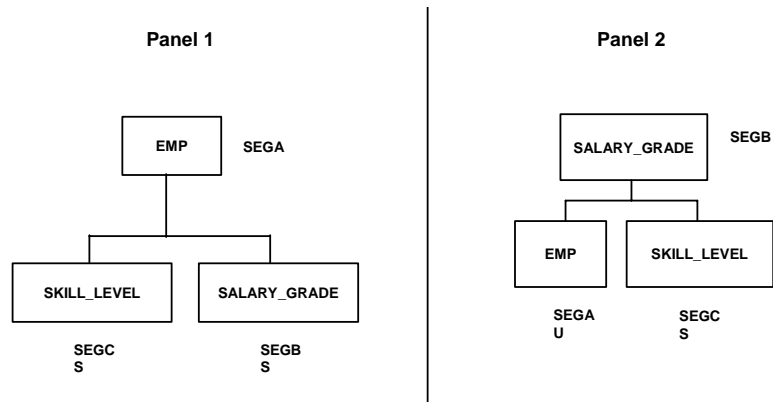
Consider this request:

```
TABLE FILE EMPFULL
LIST SKILL_LEVEL BY SALARY_GRADE
END
```

In panel 1 of the following figure, an error occurs because segments C and B are not on the same path. Therefore, you must use an inverted view:

```
TABLE FILE EMPFULL.SALARY_GRADE
LIST SKILL_LEVEL BY SALARY_GRADE
END
```

In the inverted view (panel 2), segment C is a descendant of segment B. Using this inverted view, the request can be executed.



As this request is executed, record occurrences multiply. Every record of segment C is paired with every record in segment B. If, for example, A had two B descendants and four C descendants, the report would contain eight lines of output. This effect is advantageous when it is necessary to pair every record associated with one linkpath to a record associated with another linkpath. Record pairing may produce undesirable results when the inverted segments are not directly related to each other.

If you use file inversion in conjunction with MISSING=ON, you may access orphan record occurrences that could not be accessed with the default Master File. An orphan record occurrence is one that has no parent record connection. Due to the network structure of IDMS, any hierarchical view may contain orphans. IDMS set connection options OA, OM, or MM indicate the possibility of orphans. Inversion enables the data adapter to reconstruct the IDMS relationships so that these orphans can be retrieved.

The JOIN Command

With the JOIN command, you can link two or more Master Files together for reporting purposes. The connected Master Files are treated as one structure even though they remain physically separate. Master Files stay joined for the duration of your session or until you issue the JOIN CLEAR command.

You can join the Master Files describing any of these data sources to that of your IDMS data source:

- Other IDMS data sources (SUFFIX=IDMSR)
- FOCUS
- VSAM or ISAM or QSAM
- SQL or UDB (DB2®)
- DOS/DL1 or IMS™
- CA-Datcom®/DB
- MODEL 204®
- Fixed- format sequential

A JOIN structure is implemented by matching one field that is common to both data sources. The fields on the IDMS target file can be:

- An IDMS CALC field on a network record-type.
- An indexed field (FIELDTYPE=I) on a network record-type.
- A field on an LRF record.

The fields on the host file can be:

- A virtual field located in a host Master File or created as a separate command.
- Any field.

In the Master File, the names of common fields can differ, but their field formats (ACTUAL and USAGE) must be the same.

Syntax

How to Join Two Data Sources

```
JOIN field1 [WITH rfield] IN hostfile [TAG tag1]  
TO [ALL] field2 IN crfile [TAG tag2] [AS name]  
[END]
```

where:

field1, *field2*

Are the fields common to both Master Files.

WITH *rfield1*

Use only if *field1* is a virtual field; assigns a logical home with a real field in the host file.

hostfile

Is the host Master File.

TAG *tag1*

Is a tag name of up to eight characters (usually the name of the Master File), which is used as a unique qualifier for fields and aliases in the host file.

The tag name for the host file must be the same in all the JOIN commands of a joined structure.

ALL

Use if non-unique relationships exist in the target file.

crfile

Is the target or cross-referenced Master File.

TAG *tag2*

Is a tag name of up to eight characters (usually the name of the Master File), which is used as a unique qualifier for fields and aliases in the cross-referenced file. In a recursive joined structure, if no tag name is provided, all field names and aliases are prefixed with the first four characters of the join name.

AS *name*

Assigns a name to the JOIN structure. You must assign a unique name to a join structure if:

- You want to ensure that a subsequent JOIN command will not overwrite it.
- You want to clear it selectively later.
- The structure is recursive, and you do not specify tag names.

END

Required when the JOIN command is longer than one line; terminates the command.

To join more than two files as a single structure, indicate the common fields as follows:

```
JOIN field1 IN file1 TO field2 IN file2 AS name1  
JOIN field3 IN file1 TO field4 IN file3 AS name2
```

Reference

Usage Notes for the JOIN Command

- Up to 16 joins may be active in one session.
- For a FOCUS target file, field2 must be indexed (FIELDTYPE=I).
- If you intend to use a virtual field as *field1*, specify its field name in the JOIN command and then issue its DEFINE command. Any DEFINE commands issued prior to the JOIN are cleared.
- If you know that the target file is unique, omit the ALL in the JOIN command; omitting ALL reduces I/O overhead.
- To display the JOIN structure, use the CHECK FILE command and specify the name of the host file.

Syntax

How to List JOIN Structures

To list your JOIN structures, enter:

```
? JOIN
```

Syntax

How to Clear JOIN Structures

To clear a specific JOIN structure, specify the name that you assigned to the join:

```
JOIN CLEAR name
```

Syntax

How to Clear All JOIN Structures

To clear all structures, use an asterisk (*) instead of a join name:

```
JOIN CLEAR *
```

For more information about JOIN commands, refer to your FOCUS documentation.

Example**Reporting From a Joined Structure**

This example joins the FOCUS data source JOBFIL to the IDMS EMPFULL data source based on job codes. First the JOBCODE field in JOBFIL is edited to make it compatible with the JOB_ID field in EMPFULL. The JOIN command is issued prior to the DEFINE. If the DEFINE were issued first, it would be cleared by the JOIN command:

```
JOIN JOBID WITH JOBCODE IN JOBFIL TO
ALL JOB_ID IN EMPFULL AS J1
END
DEFINE FILE JOBFIL
JCODE/A2 = IF JOBCODE LIKE 'A__' THEN '10' ELSE '20';
JOBID/A4 = JCODE|EDIT(JOBCODE,'$99');
END
TABLE FILE JOBFIL
SUM EMP_NAME IN 25 TITLE
BY DEPT_NAME
END
```

The output is:

DEPT_NAME	EMP_NAME	TITLE
-----	-----	-----
ACCOUNTING AND PAYROLL	RUPERT JENSON	MGR ACCTNG/PAYROLL
PERSONNEL	ELEANOR PEOPLES	MGR PERSONNEL

CHAPTER 8

CA-IDMS Record Retrieval Intervals

Topics:

- Entry Segment Retrieval of Network Records
- Descendant Segment Retrieval of Network Records
- Retrieving LRF-Based Records

To retrieve all of the necessary records to fulfill a request, the data adapter navigates the IDMS data source using DML or LRF commands. The data adapter automatically generates DML or LRF commands based on information from the Master File, Access File, and your request to determine the most appropriate and efficient IDMS retrieval method.

There are three kinds of IDMS access:

- Entry segment retrieval for network (DML) records.
- Descendant segment retrieval for DML records.
- LRF record retrieval.

This chapter discusses the navigational strategies used for each kind of access.

Entry Segment Retrieval of Network Records

As described in Chapter 7, *CA-IDMS Data Adapter Reporting Techniques*, a retrieval subtree is constructed based on the Master and Access Files and your request. The root of this subtree is called the entry segment because retrieval search of the data source begins at that point. The actual IDMS retrieval calls used on the entry segment depend on the entry segment's Access File information and the selection criteria in the request. To perform the most efficient record retrieval on the entry segment, the data adapter chooses one of the following techniques:

1. Retrieval by the IDMS database key (DBKEY).
2. Retrieval by the CALC field.
3. Retrieval by an index (fully-qualified, generic, or range).
4. Retrieval by area sweep.

These techniques are listed in descending order of efficiency. The idea behind selection logic is to perform as many selection tests as possible at the IDMS level. This minimizes the actual I/O operations required to access the necessary data. Area sweeps are the least desirable because they read through every record-type in the named area, including record-types that correspond to other segments, and they return every entry segment record. At this point, FOCUS selects those records that satisfy the request's test criteria and discards the rest.

Retrieval by IDMS Database Key

The IDMS database key method of retrieval takes precedence over the other alternatives because it is the most efficient. This method depends on the existence of two conditions:

- A field that corresponds to the IDMS database key (ALIAS=DBKEY) for the entry segment.
- An equality test in the request on the DBKEY field.

The equality test sets the field name of the DBKEY from the entry segment in the Master File equal to specified numeric values:

```
IF field IS value1 [ OR value2 ...OR valuen]
```

The numeric value may be up to ten characters.

You can also list the values in a separate data set allocated or FILEDEFd to an arbitrary file name, and queried with:

```
IF field IS (ddname)
```

In general, it is more efficient to produce a report with DBKEY records stored in a file.

Another method is to provide an IDMS line or page number as a value in an IF test of a Dialogue Manager procedure:

```
IF field EQ &value
```

To calculate a DBKEY value, specify the IDMS page number and the relative line number on the page in this formula:

```
DBKEY = (DB_PAGE * 255) + DB_LINE
```

DB_PAGE and DB_LINE are numeric fields with a maximum of seven and three digits respectively.

For each value specified in the IF test, the data adapter calls IDMS for the record with the DBKEY using this DML command:

```
OBTAIN record DB-KEY IS value
```

Retrieval by CALC Field

If there are no selection criteria on the DBKEY for the entry segment, the second choice is CALC access. Retrieval through the CALC key, while not as efficient as DBKEY access, takes precedence over index or area sweep retrieval.

The CALC retrieval method depends on the existence of two conditions:

- The entry segment must contain a CALC key field as specified in the Access File.
- The request must contain a fully qualified equality test on the CALC field. (If the CALC field is composed of multiple Master File fields, each field must participate in the equality test.)

The IF criteria test whether the field name of the CALC key for the segment is equal to one or more fully-qualified values:

```
IF field IS value1 [ OR value2 ...OR valuen]
```

As in the DBKEY method, you can also list values for CALC fields in a separate data set and specify that data set in the IF criteria.

For each value specified in the IF criteria, the data adapter calls IDMS with the following DML command:

```
OBTAIN CALC record
```

If the Access File indicates duplicate records (CLCDUP=Y), each DML call is followed by subsequent calls:

```
OBTAIN DUPLICATE record
```

This ensures that all appropriate records are obtained to satisfy the request.

Retrieval by IDMS Index

If there is no DBKEY or CALC key test criteria in the request, the data adapter selects the index retrieval method.

Note:

An index field must be defined with the FIELDTYPE=I attribute in the Master File and a corresponding index declaration must exist in the Access File. See Chapter 4, *CA-IDMS Data Adapter Master Files*, and Chapter 5, *CA-IDMS Data Adapter Access Files*, for details on how to describe IDMS indexes.

Index retrieval is performed two ways using:

- Selection criteria for an indexed field or GROUP in the entry segment.
- The optional parameter SEQFIELD in the Access File segment declaration.

The first method requires at least one IF test that specifies the field name of the index field. The following IF test invoke index-based retrieval:

- Fully-qualified.

```
IF TITLE IS 'PROGRAMMER/ANALYST'
```

- Partially-qualified (generic); also called masking. This applies to alphanumeric fields only.

```
IF TITLE IS 'PROGRAMMER$$$$$$$'  
IF TITLE IS 'PROGRAMMER$*'
```

- Specified as a range of values.

```
IF TITLE IS FROM 'PROGRAMMER' TO 'WORD PROCESSOR'
```

- Listed in a data set.

```
IF TITLE IS (ddname)
```

When two or more IF criteria in a request qualify an index on the entry-level segment, fully qualified retrieval takes precedence over generic; generic over range. If two IF criteria are the same type, the index in the first IF test is used. All four types of IF criteria are also supported for indexes that allow duplicate values.

If the test criteria indicate index retrieval, the data adapter issues the following DML command to IDMS:

```
OBTAIN FIRST record WITHIN setname USING value
```

where:

setname

Is the name of the index set specified in the Access File.

Then, if the Access File indicates duplicate records, the data adapter issues the following DML command for index sets:

```
OBTAIN NEXT record WITHIN setname
```

The OBTAIN NEXT call is issued until all the duplicate records are retrieved. This same NEXT call is also issued if your request contains generic or range IF criteria. It is performed once for every value or range specified in the IF criteria.

Warning:

If the IXORD parameter is improperly specified in the Access File, a range IF test may erroneously produce a report with one or no records.

When two or more IF criteria on index fields exist, indicate which index field is to be used for retrieval with file inversion syntax:

`TABLE FILE filename.fieldname`

When you use the file inversion syntax for an index located in the entry (root) segment, no file inversion takes place (you are still using the same entry segment). If you want to enforce index retrieval for a given entry segment, use the second method of retrieval involving the SEQFIELD parameter.

Using The SEQFIELD Parameter

The second method of index retrieval does not require selection criteria and yet prohibits area sweeps on entry segments. To use this method, add the optional SEQFIELD parameter to the Access File and specify the field name of the indexed field as the value of SEQFIELD.

When your request does not contain IF criteria (for DBKEY, CALC key, or another index) and a SEQFIELD is specified for the entry segment, the data adapter issues this DML command to IDMS:

`OBTAIN FIRST record WITHIN setname`

where:

`setname`

Is the IDMS set name of the indexed field (IXSET parameter) from the Access File.

Then, the data adapter issues the next command until all records connected to the index set are retrieved:

`OBTAIN NEXT record WITHIN setname`

If no sort phrase (BY or ACROSS) is specified in the request, the report is produced in ascending or descending index set order. The SEQFIELD method is recommended for indexed segments in large IDMS data sources where only a small percentage of record occurrences in a given area are the record-types defined by the segments. In such cases, IDMS resource utilization can be greatly reduced through the use of this parameter.

Warning:

If the index set connection is not mandatory/automatic and this is the entry segment, some of the records in this record-type may not be accessed. In this situation, only records that are members of the index set are returned by IDMS. If this retrieval result is undesirable, you should omit the SEQFIELD parameter.

Retrieval by Area Sweep

An area sweep is the least efficient method of entry-level retrieval, because it reads through every record in an IDMS area to return a record of a given record-type. Despite its inefficiency, an area sweep is sometimes the only method available for retrieval.

The data adapter performs an area sweep if one of the following occurs:

- No equality IF test on the DBKEY for the root exists.
- No equality IF test on the CALC key for the root exists.
- No equality or range IF criteria on an indexed field exist.
- No SEQFIELD parameter is specified.

If one of these situations occurs, the data adapter issues the following DML command to IDMS:

```
OBTAIN FIRST record WITHIN areaname
```

where:

areaname

Is the name of the IDMS data source area specified in the Access File.

Next, the data adapter issues the following command and continues to issue it until all of the records are obtained:

```
OBTAIN NEXT record WITHIN areaname
```

Descendant Segment Retrieval of Network Records

To retrieve records from a descendant segment, the data adapter's navigational strategy depends on Access File parameters and the SEGTYPE parameter in the Master File. In a few cases, the selection criteria in a request also affect the strategy.

In general, the ACCESS parameter in the Access File determines retrieval strategy because it indicates how parent/descendant relationships are implemented. There is a retrieval strategy for each kind of relationship:

- Set-based retrieval
- CALC-based retrieval
- Index-based retrieval

Set-Based Retrieval

For a set relationship (ACCESS=SET), the IDMS set is walked or searched starting with the owner to obtain related member records. Set relationships are physical ones, implemented by set pointer chains. The SEGTYPE attribute in the Master File indicates whether the descendant segment is unique or non-unique.

If the descendant segment is non-unique (SEGTYPE=S), it represents a member record-type. For non-unique descendants, the data adapter issues the following command:

```
OBTAIN NEXT record WITHIN setname
```

This command is repeated until IDMS indicates that the end of the set is reached. Then the data adapter obtains records of other descendant segments for the same parent segment or, if no other descendants exist, the next parent record is retrieved.

If the descendant segment is unique (SEGTYPE=U), it represents an owner record-type. For unique descendants, the data adapter issues this command:

```
OBTAIN OWNER WITHIN setname
```

This command is issued once, since there is one owner per set. The data adapter continues to retrieve descendant records for the same parent or retrieves the next parent record.

The KEYFLD and MULTMBR attributes in the Access File also affect retrieval for certain requests. For a sorted set, the KEYFLD parameter specifies that the set is ordered by a specified field. When the request references a field from the parent segment and has an IF test (EQ, FROM...TO, GT, GE, LT, LE) on the sort field, the data adapter sends this command to IDMS:

```
OBTAIN record WITHIN setname USING value
```

If there are duplicate records (SETDUP=Y), the data adapter issues this command:

```
OBTAIN NEXT record WITHIN setname
```

When the value of the sort field changes beyond the specified range, retrieval for that segment stops. I/O operations are minimized when the sort field value is supplied in the OBTAIN command.

Note:

The means of implementing the sorted set, the traditional method or using IDMS Integrated Indices, is transparent to the data adapter.

The MULTMBR attribute indicates an IDMS multi-member set. When it is specified, the data adapter searches for other member record-types (segments) in the Access File with the same set name. As a result, all necessary IDMS areas are activated.

CALC-Based Retrieval

CALC-based relationships (ACCESS=CLC) are performed with embedded cross-references: A field in the parent segment corresponds to the CALC field in its descendant. The data adapter uses the value from the parent's field and performs entry-level IDMS retrieval. The process of retrieving records from a descendant segment is similar to that of an entry segment. The difference is that the value supplied by the parent segment acts as the selection criteria as if the request contained explicit IF criteria.

After the data adapter retrieves the host field value (KEYFLD=*value*) from the parent segment, it calls IDMS:

```
OBTAIN CALC record
```

Then, if the descendant segment is non-unique (SEGTYPE=S), the data adapter issues the following command until all of the appropriate records are obtained to satisfy the request:

```
OBTAIN DUPLICATE record
```

Note:

IF the CLCDUP parameter does not correspond to the SEGTYPE parameter, the following error message is generated

```
(FOC919) WARNING: SAMPLE SIZE IS INSUFFICIENT FOR THIS ANALYSIS
```

A descendant segment with a CALC-based relationship (ACCESS=CLC) may act as a parent and be related to its descendants using set-, CALC-, or index-based relationships.

Index-Based Retrieval

Like CALC-based relationships, index-based relationships (ACCESS=IX) also use embedded cross-references. In index-based relationships, the field in descendant segment represents an index on the IDMS record-type. The index can be an Integrated Index. The data adapter uses the value from the parent segment and performs entry-level IDMS retrieval by walking the index set. The process of retrieving records from a descendant segment is similar to that of an entry segment. The difference is that the value supplied by the parent segment acts as the selection criteria as if the request contained explicit IF criteria.

After the data adapter retrieves the host field value (KEYFLD=*value*) from the parent segment, it calls IDMS:

```
OBTAIN FIRST record WITHIN setname USING value
```

Then, if the descendant segment is non-unique (SEGTYPE=S), the data adapter issues the following command until all indexed records with the host value are retrieved:

```
OBTAIN NEXT record WITHIN setname
```

Note:

IF the IXDUP parameter does not correspond to the SEGTYPE parameter, the data adapter generates the following error message:

```
(FOC919)  WARNING: SAMPLE SIZE IS INSUFFICIENT FOR THIS ANALYSIS
```

Only a descendant segment with an Integrated Index may act as a parent and be related to its descendants using set-, CALC-, or index-based relationships.

Retrieving LRF-Based Records

To retrieve LR records, the data adapter sends LRF calls to an access module named IDMS that invokes Logical Record Facility program.

LRF-based records are retrieved when the Access File specifies MODE=LR in the subschema declaration and the data adapter constructs an LR call to IDMS with explicit or implicit selection criteria from the request. The LRF processes the request as generated by the data adapter, selects the appropriate LR path, and constructs each flat view using the full set of selection criteria. The process is highly efficient in terms of I/O; only those records that pass the selection tests are passed back to the data adapter from IDMS.

The retrieval process for LRF records is identical to that of network record-types, but the Logical Record Facility maintains its own navigational information for the data source, selects the retrieval strategy most appropriate for a given request, and maintains its own set of currencies.

When the subschema mode is Logical Record (MODE=LR), the data adapter analyzes the request and creates an LRF command to be sent to the Logical Record Facility. The LRF command has two formats. The first format is for an entry segment without selection criteria:

```
OBTAIN NEXT record
```

The second format is for an entry with selection criteria or for a descendant segment:

```
OBTAIN NEXT record WHERE expression1 [AND expressionn]
```

As indicated by the brackets, this command is also used to pass compound selection criteria to IDMS.

IDMS processes the call and returns a record if two conditions are met:

- There is a SELECT path in the LR path-group that can process the particular request. (For entry segments without selection criteria, there must be a null SELECT clause defined for the logical record.) If there is no available SELECT path, IDMS returns error message 2041, and the data adapter terminates its processing with a FOC949 error message.

(FOC949) THE NUMBER OF SORT FIELDS EXCEEDS THE MAXIMUM:

A total of (32+number of common BY fields-number of PRINT (LIST) verbs-number of no BY verbs) BY phrases can be used in MATCH request at one time. Common BY fields are any subsequence of the largest consistent MATCH sort key set.

- There is a complete LRF record created by the path-group logic. If the selected path-group can return partial records, the data adapter processes returned records until the first partial record is returned and the LR status field is not LR-FOUND or LR-NOT-FOUND. When the data adapter encounters any other status in the LR status field, it aborts the record retrieval process and returns error messages FOC967 and FOC949. Logical records that allow retrieval of partial records should not be used with the data adapter.

(FOC967) DATA SET NOT ALLOCATED/BAD ALLOCATION

TSO IEDIT: error happened while allocating the data set to be edited. Check data set name, ddname, DYNAMNBR parameter in logon procedure.

(FOC949) THE NUMBER OF SORT FIELDS EXCEEDS THE MAXIMUM:

A total of (32+number of common BY fields-number of PRINT (LIST) verbs-number of no BY verbs) BY phrases can be used in MATCH request at one time. Common BY fields are any subsequence of the largest consistent MATCH sort key set.

The data adapter continues to call IDMS for LRF records that correspond to a segment until IDMS returns LR-NOT-FOUND in the LR status field. Then the data adapter retrieves records for other segments, or it terminates retrieval and the report is produced.

APPENDIX A

Error Messages and Troubleshooting

Topics:

- Accessing CA-IDMS Data Adapter Error Files
- Generating Error Messages Online
- Common Errors and Response Codes

This appendix contains information about finding and generating data adapter error messages and troubleshooting data adapter problems.

From time to time you may experience problems using your CA-IDMS Data Adapter. In order to address your data adapter issues and concerns specifically, please check for the following conditions:

- Can you reproduce your problem outside of IDMS in a sequential or FOCUS data source?
- Did you call the data adapter? If there was an error message in the 44xx through 45xx series, a data adapter error was produced. What is the response code?
- If no error messages were produced, ensure that the IDMS.DATA library for MVS is allocated correctly to the ddname ERRORS. This library contains the member FOCIDMS, which lists the data adapter error messages. For CMS, be sure that the FOCIDMS ERRORS file is accessible.

Accessing CA-IDMS Data Adapter Error Files

If you need to see the text or explanation for any data adapter error message, you can display it online in your FOCUS session or find it in a standard FOCUS ERRORS file.

Data Adapter messages are stored in FOCIDMS ERRORS for CMS and ERRORS.DATA(FOCIDMS) for MVS.

Generating Error Messages Online

To generate an error message online, issue the ? n query (where n is the message number) at the FOCUS command level. This query displays the message number and text along with a detailed explanation of the message, if available. For example, if you receive the message

(FOC887) THE DDNAME CANNOT BE USED IN THIS CONTEXT

you can enter

? 887

to get this explanation:

(FOC887) THE DDNAME CANNOT BE USED IN THIS CONTEXT

The ddname mentioned is probably SYSPRINT, which cannot be used as a LOG file or TYPE file, since it conflicts with regular FOCUS terminal output. Use a different ddname.

Common Errors and Response Codes

The following chart lists common FOCUS error messages and IDMS response codes.

IDMS Abend Message	Description
D002	A DC##### error was issued with a severity code of 3, 4, or 5 causing the IDMS system to terminate the task.
D003	A program check has occurred during run-unit processing. Probable storage overwritten.
D004	Run away task that has been terminated by the IDMS system because it executed longer than the time specified in the SYSGEN “RUNAWAY INTERVAL” parameter without an intervening wait.
D005	IDMS/DC program has invoked an invalid CA-IDMS function or specified an invalid address for a valid function.
D009	Task terminated by the IDMS system because it exceeded its registered stack space.

IDMS Abend Message	Description
11## 30##	Usually indicates a corrupt data source or problem left behind by a DBA change.
3005	An attempt to open a data source file has failed. Check the JCL for a missing or invalid DD command or a mismatch between the DMCL page size and the disk data set blocksize.
3010	An attempt to read from the data source or disk journal has failed: 1) a physical I/O error, 2) attempt to read a relative block outside of the range, 3) failed attempt to initialize the file, 4) with VSAM, insufficient space in the storage pool.
ABRU	An external request unit executing under SYSCTL has abended. The abend may have been caused when the request unit exceeded the maximum time allowed between data source requests, or it may have been caused by other factors, as described in the system log. If the abend condition was detected by the check-user, lookaround-time, or external-wait mechanism, the system writes a SNAP DUMP to the log file.
CKUR	The CheckUser task has detected that the program ABENDED.
DC001003	Program ABENDED because it exceeded a wait time on a resource, such as a lock (a DBKEY, page lock, or area lock).

Common Errors Correctable by Database Administration

Major	Minor	Description	Action
09	66	Data source offline for a reason.	DBA may be doing updates to or maintenance on an IDMS area.
nn	11	Insufficient space.	Time must be scheduled to expand the data source.
03 nn	01 01	Required area not readied.	Relates to separating an index out from the data area and the new index area was not added to the program, or the program did not recompile properly.
09	71	Page range doesn't match DMCL.	DMCL or segment problem.

Common Errors Correctable by Operations

Major	Minor	Description	Action
nn	69	Run Unit could not start or contention problem.	Rerun job. Check CV log for abend in module IDMSDBMS or other critical system module.
14	69	CV not active.	Bring up CV and rerun job. Also, check the SYSCTL DD card and the rest of the JCL to make sure you are pointing to the correct CV.
09	66	Area not available in desired usage mode. Look at SYSOUT to find problem area-name.	Check to see if any batch local job is running. If so, wait until local job finishes and re-submit.
nn	29	Deadlock caused by two or more programs locking the same records.	Allow one of the programs to finish, then rerun the program that failed.

APPENDIX B

CA-IDMS Data Adapter Samples

Topics:

- Schema Sample: EMPSCHM
- Network Subschema Sample: EMPSS01
- Master File for Network Sample
- Access File for Network Sample
- Master File for LRF Sample
- Access File for LRF Sample
- Sample of a Partial LRF Record
- EMPFULL Master File
- EMPFULL Access File
- EMPFILE Master File
- EMPFILE Access File
- Sample CA-IDMS Report

This appendix contains:

- The schema EMPSCHM.
- A network subschema for schema EMPSCHM, plus corresponding Master and Access Files.
- A sample of a NULL SELECT command that creates an LRF partial record occurrence.
- The EMPFULL Master File and Access File, which are used in the examples throughout this manual.
- The EMPFILE Master File and Access File, which are used in the trace examples in Appendix C, *Tracing CA-IDMS Data Adapter Processing*.
- A sample report on data elements.

Note:

Some samples are annotated to illustrate a specific command.

Schema Sample: EMPSCHM

This schema is the physical description of the IDMS EMPSCHM data source. It contains the following annotated items:

1. Area-to-file and area-to-ddname mapping.
2. A CALC-based record.
3. A VIA-based record.
4. A sorted set ordered by the SKILL-LEVEL field.
5. A sorted set ordered by Integrated Indexes using the SKILL-LEVEL field.
6. An Integrated Index set ordered by the SKILL-NAME field.

```
000001      DIS  SCHEMA EMPSCHM V 100 WITHO ELE.
*+  ADD
*+  SCHEMA NAME IS EMPSCHM VERSION IS 100
*+      DATE CREATED IS      08/20/97
*+      TIME CREATED IS      15003144
*+      PREPARED BY CSDPJC
*+      SCHEMA DESCRIPTION IS 'EMPLOYEE DEMO DATABASE'
*+      ASSIGN RECORD IDS FROM 1001
*+      PUBLIC ACCESS IS ALLOWED FOR ALL
*+      COMMENTS
*+          'INSTALLATION: COMMONWEATHER CORPORATION'
*+      SUBSCHEMA IS EMPSS01
*+      .
*+  ADD
*+      AREA NAME IS EMP-DEMO-REGION
*+      ESTIMATED PAGES ARE 0
*+      .
*+  ADD
*+      AREA NAME IS ORG-DEMO-REGION
*+      ESTIMATED PAGES ARE 0
*+      .
*+  ADD
*+      AREA NAME IS INS-DEMO-REGION
*+      ESTIMATED PAGES ARE 0
*+      .
```

```

*+      ADD
*+      RECORD NAME IS COVERAGE
*+          SHARE STRUCTURE OF RECORD COVERAGE VERSION 100
*+          RECORD ID IS 400
*+          LOCATION MODE IS VIA EMP-COVERAGE SET
*+          WITHIN AREA INS-DEMO-REGION OFFSET 5 PAGES FOR 45 PAGES
*+          RECORD NAME SYNONYM IS COVERGE FOR LANGUAGE ASSEMBLER
*+          RECORD NAME SYNONYM IS COVRGE FOR LANGUAGE FORTRAN
*+          OWNER OF SET COVERAGE-CLAIMS
*+              NEXT DBKEY POSITION IS 4
*+              PRIOR DBKEY POSITION IS 5
*+          MEMBER OF SET EMP-COVERAGE
*+              NEXT DBKEY POSITION IS 1
*+              PRIOR DBKEY POSITION IS 2
*+              OWNER DBKEY POSITION IS 3
*+      .
*+      ADD
*+      RECORD NAME IS DENTAL-CLAIM
*+          SHARE STRUCTURE OF RECORD DENTAL-CLAIM VERSION 100
*+          RECORD ID IS 405
*+          LOCATION MODE IS VIA COVERAGE-CLAIMS SET
*+          MINIMUM ROOT LENGTH IS 132 CHARACTERS
*+          MINIMUM FRAGMENT LENGTH IS 932 CHARACTERS
*+          WITHIN AREA INS-DEMO-REGION OFFSET 5 PAGES FOR 45 PAGES
*+          RECORD NAME SYNONYM IS DENTCLM FOR LANGUAGE ASSEMBLER
*+          RECORD NAME SYNONYM IS DNTCLM FOR LANGUAGE FORTRAN
*+          MEMBER OF SET COVERAGE-CLAIMS
*+              NEXT DBKEY POSITION IS 1
*+              PRIOR DBKEY POSITION IS 2
*+      .
*+      ADD
*+      RECORD NAME IS DEPARTMENT
*+          SHARE STRUCTURE OF RECORD DEPARTMENT VERSION 100
*+          RECORD ID IS 410
*+          LOCATION MODE IS CALC USING ( DEPT-ID-0410 )
*+              DUPLICATES ARE NOT ALLOWED
*+          WITHIN AREA ORG-DEMO-REGION OFFSET 5 PAGES FOR 45 PAGES
*+          RECORD NAME SYNONYM IS DEPARTMT FOR LANGUAGE ASSEMBLER
*+          RECORD NAME SYNONYM IS DEPT FOR LANGUAGE FORTRAN
*+          OWNER OF SET DEPT-EMPLOYEE
*+              NEXT DBKEY POSITION IS 1
*+              PRIOR DBKEY POSITION IS 2
*+      .
*+      ADD
*+      RECORD NAME IS EMPLOYEE
*+          SHARE STRUCTURE OF RECORD EMPLOYEE VERSION 100
*+          RECORD ID IS 415
*+          LOCATION MODE IS CALC USING ( EMP-ID-0415 )
*+              DUPLICATES ARE NOT ALLOWED
*+          WITHIN AREA EMP-DEMO-REGION OFFSET 5 PAGES FOR 95 PAGES
*+          RECORD NAME SYNONYM IS EMPLOYE FOR LANGUAGE ASSEMBLER
*+          RECORD NAME SYNONYM IS EMPLOY FOR LANGUAGE FORTRAN
*+          OWNER OF SET EMP-COVERAGE

```

```

*+          NEXT DBKEY POSITION IS 7
*+          PRIOR DBKEY POSITION IS 8
*+      OWNER OF SET EMP-EMPOSITION
*+          NEXT DBKEY POSITION IS 9
*+          PRIOR DBKEY POSITION IS 10
*+      OWNER OF SET EMP-EXPERTISE
*+          NEXT DBKEY POSITION IS 11
*+          PRIOR DBKEY POSITION IS 12
*+      OWNER OF SET MANAGES
*+          NEXT DBKEY POSITION IS 13
*+          PRIOR DBKEY POSITION IS 14
*+      OWNER OF SET REPORTS-TO
*+          NEXT DBKEY POSITION IS 15
*+          PRIOR DBKEY POSITION IS 16
*+      MEMBER OF SET DEPT-EMPLOYEE
*+          NEXT DBKEY POSITION IS 1
*+          PRIOR DBKEY POSITION IS 2
*+          OWNER DBKEY POSITION IS 3
*+      MEMBER OF SET EMP-NAME-NDX
*+          INDEX DBKEY POSITION IS 4
*+      MEMBER OF SET OFFICE-EMPLOYEE
*+          INDEX DBKEY POSITION IS 5
*+          OWNER DBKEY POSITION IS 6
*+      .
*+      ADD
*+      RECORD NAME IS EMPOSITION
*+          SHARE STRUCTURE OF RECORD EMPOSITION VERSION 100
*+          RECORD ID IS 420
*+          LOCATION MODE IS VIA EMP-EMPOSITION SET
*+          WITHIN AREA EMP-DEMO-REGION OFFSET 5 PAGES FOR 95 PAGES
*+          RECORD NAME SYNONYM IS EMPOSITN FOR LANGUAGE ASSEMBLER
*+          RECORD NAME SYNONYM IS EMPOST FOR LANGUAGE FORTRAN
*+      MEMBER OF SET EMP-EMPOSITION
*+          NEXT DBKEY POSITION IS 1
*+          PRIOR DBKEY POSITION IS 2
*+          OWNER DBKEY POSITION IS 3
*+      MEMBER OF SET JOB-EMPOSITION
*+          NEXT DBKEY POSITION IS 4
*+          PRIOR DBKEY POSITION IS 5
*+          OWNER DBKEY POSITION IS 6
*+      .
*+      ADD
*+      RECORD NAME IS EXPERTISE
*+          SHARE STRUCTURE OF RECORD EXPERTISE VERSION 100
*+          RECORD ID IS 425
*+          LOCATION MODE IS VIA EMP-EXPERTISE SET
*+          WITHIN AREA EMP-DEMO-REGION OFFSET 5 PAGES FOR 95 PAGES
*+          RECORD NAME SYNONYM IS EXPRTISE FOR LANGUAGE ASSEMBLER
*+          RECORD NAME SYNONYM IS EXPRTS FOR LANGUAGE FORTRAN
*+      MEMBER OF SET EMP-EXPERTISE
*+          NEXT DBKEY POSITION IS 1
*+          PRIOR DBKEY POSITION IS 2
*+          OWNER DBKEY POSITION IS 3

```

```

*+      MEMBER OF SET SKILL-EXPERTISE
*+      INDEX DBKEY POSITION IS 4
*+      OWNER DBKEY POSITION IS 5
*+      .
*+  ADD
*+  RECORD NAME IS HOSPITAL-CLAIM
*+      SHARE STRUCTURE OF RECORD HOSPITAL-CLAIM VERSION 100
*+      RECORD ID IS 430
*+      LOCATION MODE IS VIA COVERAGE-CLAIMS SET
*+      WITHIN AREA INS-DEMO-REGION OFFSET 5 PAGES FOR 45 PAGES
*+      RECORD NAME SYNONYM IS HOSPCLM FOR LANGUAGE ASSEMBLER
*+      RECORD NAME SYNONYM IS HSPCLM FOR LANGUAGE FORTRAN
*+      MEMBER OF SET COVERAGE-CLAIMS
*+      NEXT DBKEY POSITION IS 1
*+      PRIOR DBKEY POSITION IS 2
*+      .
*+  ADD
*+  RECORD NAME IS INSURANCE-PLAN
*+      SHARE STRUCTURE OF RECORD INSURANCE-PLAN VERSION 100
*+      RECORD ID IS 435
*+      LOCATION MODE IS CALC USING ( INS-PLAN-CODE-0435 )
*+      DUPLICATES ARE NOT ALLOWED
*+      WITHIN AREA INS-DEMO-REGION OFFSET 1 PAGES FOR 4 PAGES
*+      RECORD NAME SYNONYM IS INSPLAN FOR LANGUAGE ASSEMBLER
*+      RECORD NAME SYNONYM IS INSPLN FOR LANGUAGE FORTRAN
*+      .
*+  ADD
*+  RECORD NAME IS JOB
*+      SHARE STRUCTURE OF RECORD JOB VERSION 100
*+      RECORD ID IS 440
*+      LOCATION MODE IS CALC USING ( JOB-ID-0440 )
*+      DUPLICATES ARE NOT ALLOWED
*+      MINIMUM ROOT LENGTH IS 24 CHARACTERS
*+      MINIMUM FRAGMENT LENGTH IS 296 CHARACTERS
*+      CALL IDMSCOMP BEFORE STORE
*+      CALL IDMSCOMP BEFORE MODIFY
*+      CALL IDMSDCOM AFTER GET
*+      WITHIN AREA ORG-DEMO-REGION OFFSET 5 PAGES FOR 45 PAGES
*+      RECORD NAME SYNONYM IS JOBA FOR LANGUAGE ASSEMBLER
*+      RECORD NAME SYNONYM IS JOBF FOR LANGUAGE FORTRAN
*+      OWNER OF SET JOB-EMPOSITION
*+      NEXT DBKEY POSITION IS 2
*+      PRIOR DBKEY POSITION IS 3
*+      MEMBER OF SET JOB-TITLE-NDX
*+      INDEX DBKEY POSITION IS 1
*+      .
*+  ADD
*+  RECORD NAME IS NON-HOSP-CLAIM
*+      SHARE STRUCTURE OF RECORD NON-HOSP-CLAIM VERSION 100
*+      RECORD ID IS 445
*+      LOCATION MODE IS VIA COVERAGE-CLAIMS SET
*+      MINIMUM ROOT LENGTH IS 248 CHARACTERS
*+      MINIMUM FRAGMENT LENGTH IS 1052 CHARACTERS

```

```

*+      WITHIN AREA INS-DEMO-REGION OFFSET 5 PAGES FOR 45 PAGES
*+      RECORD NAME SYNONYM IS NONHSPCL FOR LANGUAGE ASSEMBLER
*+      RECORD NAME SYNONYM IS NHSPCL FOR LANGUAGE FORTRAN
*+      MEMBER OF SET COVERAGE-CLAIMS
*+      NEXT DBKEY POSITION IS 1
*+      PRIOR DBKEY POSITION IS 2
*+      .
*+  ADD
*+  RECORD NAME IS OFFICE
*+      SHARE STRUCTURE OF RECORD OFFICE VERSION 100
*+      RECORD ID IS 450
*+      LOCATION MODE IS CALC USING ( OFFICE-CODE-0450 )
*+      DUPLICATES ARE NOT ALLOWED
*+      WITHIN AREA ORG-DEMO-REGION OFFSET 5 PAGES FOR 45 PAGES
*+      RECORD NAME SYNONYM IS OFFIC FOR LANGUAGE ASSEMBLER
*+      RECORD NAME SYNONYM IS OFFCE FOR LANGUAGE FORTRAN
*+      OWNER OF SET OFFICE-EMPLOYEE
*+      NEXT DBKEY POSITION IS 1
*+      PRIOR DBKEY POSITION IS 2
*+      .
*+  ADD
*+  RECORD NAME IS SKILL
*+      SHARE STRUCTURE OF RECORD SKILL VERSION 100
*+      RECORD ID IS 455
*+      LOCATION MODE IS CALC USING ( SKILL-ID-0455 )
*+      DUPLICATES ARE NOT ALLOWED
*+      WITHIN AREA ORG-DEMO-REGION OFFSET 5 PAGES FOR 45 PAGES
*+      RECORD NAME SYNONYM IS SKILLA FOR LANGUAGE ASSEMBLER
*+      RECORD NAME SYNONYM IS SKILF FOR LANGUAGE FORTRAN
*+      OWNER OF SET SKILL-EXPERTISE
*+      NEXT DBKEY POSITION IS 2
*+      PRIOR DBKEY POSITION IS 3
*+      MEMBER OF SET SKILL-NAME-NDX
*+      INDEX DBKEY POSITION IS 1
*+      .
*+  ADD
*+  RECORD NAME IS STRUCTURE
*+      SHARE STRUCTURE OF RECORD STRUCTURE VERSION 100
*+      RECORD ID IS 460
*+      LOCATION MODE IS VIA MANAGES SET
*+      WITHIN AREA EMP-DEMO-REGION OFFSET 5 PAGES FOR 95 PAGES
*+      RECORD NAME SYNONYM IS STRUCTUR FOR LANGUAGE ASSEMBLER
*+      RECORD NAME SYNONYM IS STRUCT FOR LANGUAGE FORTRAN
*+      MEMBER OF SET MANAGES
*+      NEXT DBKEY POSITION IS 1
*+      PRIOR DBKEY POSITION IS 2
*+      OWNER DBKEY POSITION IS 3
*+      MEMBER OF SET REPORTS-TO
*+      NEXT DBKEY POSITION IS 4
*+      PRIOR DBKEY POSITION IS 5
*+      OWNER DBKEY POSITION IS 6
*+      .
*+  ADD

```

```

*+ SET NAME IS COVERAGE-CLAIMS
*+ ORDER IS LAST
*+ MODE IS CHAIN LINKED TO PRIOR
*+ OWNER IS COVERAGE
*+ WITHIN AREA INS-DEMO-REGION
*+ NEXT DBKEY POSITION IS 4
*+ PRIOR DBKEY POSITION IS 5
*+ MEMBER IS HOSPITAL-CLAIM
*+ WITHIN AREA INS-DEMO-REGION
*+ NEXT DBKEY POSITION IS 1
*+ PRIOR DBKEY POSITION IS 2
*+ MANDATORY AUTOMATIC
*+ MEMBER IS NON-HOSP-CLAIM
*+ WITHIN AREA INS-DEMO-REGION
*+ NEXT DBKEY POSITION IS 1
*+ PRIOR DBKEY POSITION IS 2
*+ MANDATORY AUTOMATIC
*+ MEMBER IS DENTAL-CLAIM
*+ WITHIN AREA INS-DEMO-REGION
*+ NEXT DBKEY POSITION IS 1
*+ PRIOR DBKEY POSITION IS 2
*+ MANDATORY AUTOMATIC
*+ .
*+ ADD
*+ SET NAME IS DEPT-EMPLOYEE
*+ ORDER IS SORTED
*+ MODE IS CHAIN LINKED TO PRIOR
*+ OWNER IS DEPARTMENT
*+ WITHIN AREA ORG-DEMO-REGION
*+ NEXT DBKEY POSITION IS 1
*+ PRIOR DBKEY POSITION IS 2
*+ MEMBER IS EMPLOYEE
*+ WITHIN AREA EMP-DEMO-REGION
*+ NEXT DBKEY POSITION IS 1
*+ PRIOR DBKEY POSITION IS 2
*+ LINKED TO OWNER
*+ OWNER DBKEY POSITION IS 3
*+ OPTIONAL AUTOMATIC
*+ KEY IS (
*+ EMP-LAST-NAME-0415 ASCENDING
*+ EMP-FIRST-NAME-0415 ASCENDING )
*+ DUPLICATES ARE LAST
*+ NATURAL SEQUENCE
*+ .
*+ ADD
*+ SET NAME IS EMP-COVERAGE
*+ ORDER IS FIRST
*+ MODE IS CHAIN LINKED TO PRIOR
*+ OWNER IS EMPLOYEE
*+ WITHIN AREA EMP-DEMO-REGION
*+ NEXT DBKEY POSITION IS 7
*+ PRIOR DBKEY POSITION IS 8
*+ MEMBER IS COVERAGE

```



```

*+          WITHIN AREA INS-DEMO-REGION
*+          NEXT DBKEY POSITION IS 1
*+          PRIOR DBKEY POSITION IS 2
*+          LINKED TO OWNER
*+          OWNER DBKEY POSITION IS 3
*+          MANDATORY AUTOMATIC
*+
*+      .
*+      ADD
*+      SET NAME IS EMP-EMPOSITION
*+          ORDER IS FIRST
*+          MODE IS CHAIN LINKED TO PRIOR
*+          OWNER IS EMPLOYEE
*+          WITHIN AREA EMP-DEMO-REGION
*+          NEXT DBKEY POSITION IS 9
*+          PRIOR DBKEY POSITION IS 10
*+          MEMBER IS EMPOSITION
*+          WITHIN AREA EMP-DEMO-REGION
*+          NEXT DBKEY POSITION IS 1
*+          PRIOR DBKEY POSITION IS 2
*+          LINKED TO OWNER
*+          OWNER DBKEY POSITION IS 3
*+          MANDATORY AUTOMATIC
*+
*+      .
*+      ADD
*+      SET NAME IS EMP-EXPERTISE
*+          ORDER IS SORTED
*+          MODE IS CHAIN LINKED TO PRIOR
*+          OWNER IS EMPLOYEE
*+          WITHIN AREA EMP-DEMO-REGION
*+          NEXT DBKEY POSITION IS 11
*+          PRIOR DBKEY POSITION IS 12
*+          MEMBER IS EXPERTISE
*+          WITHIN AREA EMP-DEMO-REGION
*+          NEXT DBKEY POSITION IS 1
*+          PRIOR DBKEY POSITION IS 2
*+          LINKED TO OWNER
*+          OWNER DBKEY POSITION IS 3
*+          MANDATORY AUTOMATIC
*+          KEY IS (
*+              SKILL-LEVEL-0425 DESCENDING )
*+              DUPLICATES ARE FIRST
*+              NATURAL SEQUENCE
*+
*+      .
*+      ADD
*+      SET NAME IS EMP-NAME-NDX
*+          ORDER IS SORTED
*+          MODE IS INDEX BLOCK CONTAINS 40 KEYS
*+          OWNER IS SYSTEM
*+          WITHIN AREA EMP-DEMO-REGION OFFSET 1 PAGES FOR 4 PAGES
*+          MEMBER IS EMPLOYEE
*+          WITHIN AREA EMP-DEMO-REGION
*+          INDEX DBKEY POSITION IS 4
*+          OPTIONAL AUTOMATIC

```

```

*+          KEY IS (
*+              EMP-LAST-NAME-0415 ASCENDING
*+              EMP-FIRST-NAME-0415 ASCENDING )
*+              DUPLICATES ARE LAST
*+              NATURAL SEQUENCE
*+              COMPRESSED
*+      .
*+  ADD
*+  SET NAME IS JOB-EMPOSITION
*+      ORDER IS NEXT
*+      MODE IS CHAIN LINKED TO PRIOR
*+      OWNER IS JOB
*+          WITHIN AREA ORG-DEMO-REGION
*+          NEXT DBKEY POSITION IS 2
*+          PRIOR DBKEY POSITION IS 3
*+      MEMBER IS EMPOSITION
*+          WITHIN AREA EMP-DEMO-REGION
*+          NEXT DBKEY POSITION IS 4
*+          PRIOR DBKEY POSITION IS 5
*+          LINKED TO OWNER
*+          OWNER DBKEY POSITION IS 6
*+          OPTIONAL MANUAL
*+      .
*+  ADD
*+  SET NAME IS JOB-TITLE-NDX
*+      ORDER IS SORTED
*+      MODE IS INDEX BLOCK CONTAINS 30 KEYS
*+      OWNER IS SYSTEM
*+          WITHIN AREA ORG-DEMO-REGION OFFSET 1 PAGES FOR 4 PAGES
*+      MEMBER IS JOB
*+          WITHIN AREA ORG-DEMO-REGION
*+          INDEX DBKEY POSITION IS 1
*+          OPTIONAL AUTOMATIC
*+      KEY IS (
*+          TITLE-0440 ASCENDING )
*+          DUPLICATES ARE NOT ALLOWED
*+          NATURAL SEQUENCE
*+          UNCOMPRESSED
*+      .
*+  ADD
*+  SET NAME IS MANAGES
*+      ORDER IS NEXT
*+      MODE IS CHAIN LINKED TO PRIOR
*+      OWNER IS EMPLOYEE
*+          WITHIN AREA EMP-DEMO-REGION
*+          NEXT DBKEY POSITION IS 13
*+          PRIOR DBKEY POSITION IS 14
*+      MEMBER IS STRUCTURE
*+          WITHIN AREA EMP-DEMO-REGION
*+          NEXT DBKEY POSITION IS 1
*+          PRIOR DBKEY POSITION IS 2
*+          LINKED TO OWNER
*+          OWNER DBKEY POSITION IS 3

```

```

*+          MANDATORY AUTOMATIC
*+
*+      .
*+      ADD
*+      SET NAME IS OFFICE-EMPLOYEE
*+          ORDER IS SORTED
*+          MODE IS INDEX BLOCK CONTAINS 30 KEYS
*+          OWNER IS OFFICE
*+              WITHIN AREA ORG-DEMO-REGION
*+              NEXT DBKEY POSITION IS 1
*+              PRIOR DBKEY POSITION IS 2
*+      MEMBER IS EMPLOYEE
*+          WITHIN AREA EMP-DEMO-REGION
*+          INDEX DBKEY POSITION IS 5
*+          LINKED TO OWNER
*+          OWNER DBKEY POSITION IS 6
*+          OPTIONAL AUTOMATIC
*+          KEY IS (
*+              EMP-LAST-NAME-0415 ASCENDING
*+              EMP-FIRST-NAME-0415 ASCENDING )
*+              DUPLICATES ARE LAST
*+              NATURAL SEQUENCE
*+              COMPRESSED
*+      .
*+      ADD
*+      SET NAME IS REPORTS-TO
*+          ORDER IS NEXT
*+          MODE IS CHAIN LINKED TO PRIOR
*+          OWNER IS EMPLOYEE
*+              WITHIN AREA EMP-DEMO-REGION
*+              NEXT DBKEY POSITION IS 15
*+              PRIOR DBKEY POSITION IS 16
*+      MEMBER IS STRUCTURE
*+          WITHIN AREA EMP-DEMO-REGION
*+          NEXT DBKEY POSITION IS 4
*+          PRIOR DBKEY POSITION IS 5
*+          LINKED TO OWNER
*+          OWNER DBKEY POSITION IS 6
*+          OPTIONAL MANUAL
*+      .
*+      ADD
*+      SET NAME IS SKILL-EXPERTISE
*+          ORDER IS SORTED
*+          MODE IS INDEX BLOCK CONTAINS 30 KEYS
*+          OWNER IS SKILL
*+              WITHIN AREA ORG-DEMO-REGION
*+              NEXT DBKEY POSITION IS 2
*+              PRIOR DBKEY POSITION IS 3
*+      MEMBER IS EXPERTISE
*+          WITHIN AREA EMP-DEMO-REGION
*+          INDEX DBKEY POSITION IS 4
*+          LINKED TO OWNER
*+          OWNER DBKEY POSITION IS 5
*+          MANDATORY AUTOMATIC

```

```

*+          KEY IS (
*+             SKILL-LEVEL-0425 DESCENDING )
*+             DUPLICATES ARE FIRST
*+             NATURAL SEQUENCE
*+             UNCOMPRESSED
*+         .
*+     ADD
*+     SET NAME IS SKILL-NAME-NDX
*+     ORDER IS SORTED
*+     MODE IS INDEX BLOCK CONTAINS 30 KEYS
*+     OWNER IS SYSTEM
*+     WITHIN AREA ORG-DEMO-REGION OFFSET 1 PAGES FOR 4 PAGES
*+     MEMBER IS SKILL
*+     WITHIN AREA ORG-DEMO-REGION
*+     INDEX DBKEY POSITION IS 1
*+     OPTIONAL AUTOMATIC
*+     KEY IS (
*+         SKILL-NAME-0455 ASCENDING )
*+         DUPLICATES ARE NOT ALLOWED
*+         NATURAL SEQUENCE
*+         UNCOMPRESSED
*+     .
*+
** TRANSACTION SUMMARY **
ENTITY          ADD MODIFY REPLACE DELETE DISPLAY
.....
SCHEMA          0      0      0      0      1

NO ERRORS OR WARNINGS ISSUED FOR THIS COMPILE

```

Network Subschema Sample: EMPSS01

This subschema shows the network view of schema EMPSCHEM. See Chapter 3, *IDMS Overview and Mapping Concepts*, for its Entity Relationship diagram.

```

000001          DISPLAY SUB EMPSS01 OF SCHEMA EMPSCHEM V 100 WITHO ELE.
*+     ADD
*+     SUBSCHEMA NAME IS EMPSS01 OF SCHEMA NAME IS EMPSCHEM VERSION IS 100
*+         DATE CREATED IS          08/20/97
*+         TIME CREATED IS          15003500
*+         DATE LAST UPDATED IS    08/31/00
*+         TIME LAST UPDATED IS    11471506
*+         PREPARED BY CSDPJC
*+         REVISED  BY PMSEXW
*+         USER IS PMSEXW
*+         REGISTERED FOR PUBLIC ACCESS
*+         RESPONSIBLE FOR NONE
*+         PUBLIC ACCESS IS ALLOWED FOR ALL
*+         USAGE IS MIXED
*+         COMMENTS
*+         'THIS IS THE COMPLETE VIEW OF EMPSCHEM'

```

```
*+ .
*+ ADD
*+ AREA NAME IS EMP-DEMO-REGION
*+ .
*+ ADD
*+ AREA NAME IS INS-DEMO-REGION
*+ .
*+ ADD
*+ AREA NAME IS ORG-DEMO-REGION
*+ .
*+ ADD
*+ RECORD NAME IS COVERAGE
*+ .
*+ ADD
*+ RECORD NAME IS DENTAL-CLAIM
*+ .
*+ ADD
*+ RECORD NAME IS DEPARTMENT
*+ .
*+ ADD
*+ RECORD NAME IS EMPLOYEE
*+ .
*+ ADD
*+ RECORD NAME IS EMPOSITION
*+ .
*+ ADD
*+ RECORD NAME IS EXPERTISE
*+ .
*+ ADD
*+ RECORD NAME IS HOSPITAL-CLAIM
*+ .
*+ ADD
*+ RECORD NAME IS INSURANCE-PLAN
*+ .
*+ ADD
*+ RECORD NAME IS JOB
*+ .
*+ ADD
*+ RECORD NAME IS NON-HOSP-CLAIM
*+ .
*+ ADD
*+ RECORD NAME IS OFFICE
*+ .
*+ ADD
*+ RECORD NAME IS SKILL
*+ .
*+ ADD
*+ RECORD NAME IS STRUCTURE
*+ .
*+ ADD
*+ SET NAME IS COVERAGE-CLAIMS
*+ .
*+ ADD
```

```

**+   SET NAME IS DEPT-EMPLOYEE
**+   .
**+   ADD
**+   SET NAME IS EMP-COVERAGE
**+   .
**+   ADD
**+   SET NAME IS EMP-EXPERTISE
**+   .
**+   ADD
**+   SET NAME IS EMP-NAME-NDX
**+   .
**+   ADD
**+   SET NAME IS EMP-EMPOSITION
**+   .
**+   ADD
**+   SET NAME IS JOB-EMPOSITION
**+   .
**+   ADD
**+   SET NAME IS JOB-TITLE-NDX
**+   .
**+   ADD
**+   SET NAME IS MANAGES
**+   .
**+   ADD
**+   SET NAME IS OFFICE-EMPLOYEE
**+   .
**+   ADD
**+   SET NAME IS REPORTS-TO
**+   .
**+   ADD
**+   SET NAME IS SKILL-EXPERTISE
**+   .
**+   ADD
**+   SET NAME IS SKILL-NAME-NDX
**+   .

          ** TRANSACTION SUMMARY **
ENTITY              ADD MODIFY REPLACE DELETE DISPLAY
.....

SUBSCHEMA              0      0      0      0      1

NO ERRORS OR WARNINGS ISSUED FOR THIS COMPILE

```

Master File for Network Sample

This Master File corresponds to network subschema EMPSS01:

```

FILE=EMPNETWK,SUFFIX=IDMSR ,,$

SEGNAME=DEPT,$
  FIELDNAME=DEPT_ID      ,ALIAS=DID      ,USAGE=A4      ,ACTUAL=A4      ,,$
  FIELDNAME=DEPT_NAME    ,ALIAS=DNAME    ,USAGE=A45     ,ACTUAL=A45     ,,$
  FIELDNAME=DEPT_HEAD    ,ALIAS=DHEAD    ,USAGE=A4      ,ACTUAL=A4      ,,$
  FIELDNAME=DEPT_DBKEY    ,ALIAS=DBKEY    ,USAGE=I10     ,ACTUAL=I4      ,,$

SEGNAME=EMPLOYE,PARENT=DEPT,SEGTYPE=S,$
  FIELDNAME=EMP_ID      ,ALIAS=EID      ,USAGE=A4      ,ACTUAL=A4      ,,$
  GROUP=EMP_NAME        ,ALIAS=ENAME    ,USAGE=A25     ,ACTUAL=A25     ,,$
    FIELDNAME=FIRST_NAME,ALIAS=EFN      ,USAGE=A10     ,ACTUAL=A10     ,,$
    FIELDNAME=LAST_NAME ,ALIAS=ELN      ,USAGE=A15     ,ACTUAL=A15     ,,$
  FIELDNAME=EMP_STREET  ,ALIAS=ESTR     ,USAGE=A20     ,ACTUAL=A20     ,,$
  FIELDNAME=EMP_CITY    ,ALIAS=ECI      ,USAGE=A15     ,ACTUAL=A15     ,,$
  FIELDNAME=EMP_STATE   ,ALIAS=EST      ,USAGE=A2      ,ACTUAL=A2      ,,$
  GROUP=EMP_FULL_ZIP    ,ALIAS=EFZIP    ,USAGE=A9      ,ACTUAL=A9      ,,$
    FIELDNAME=EMP_ZIP    ,ALIAS=EZIP     ,USAGE=A5      ,ACTUAL=A5      ,,$
    FIELDNAME=EMP_ZIP_L  ,ALIAS=EZIPL    ,USAGE=A4      ,ACTUAL=A4      ,,$
  FIELDNAME=EMP_PHONE    ,ALIAS=EPHON    ,USAGE=A10     ,ACTUAL=A10     ,,$
  FIELDNAME=STATUS      ,ALIAS=ESTAT    ,USAGE=A2      ,ACTUAL=A2      ,,$
  FIELDNAME=SOC_SEC_NUM ,ALIAS=SSN      ,USAGE=A9      ,ACTUAL=A9      ,,$
  FIELDNAME=EMP_STRT_DTE,ALIAS=ESDTE    ,USAGE=A6YMD   ,ACTUAL=A6      ,,$
  FIELDNAME=EMP_TERM_DTE,ALIAS=ETDTE    ,USAGE=A6YMD   ,ACTUAL=A6      ,,$
  FIELDNAME=EMP_BRTH_DTE,ALIAS=EBDTE    ,USAGE=A6YMD   ,ACTUAL=A6      ,,$
  FIELDNAME=EMP_DBKEY    ,ALIAS=DBKEY    ,USAGE=I10     ,ACTUAL=I4      ,,$

SEGNAME=OFFICE,PARENT=EMPLOYE,SEGTYPE=U,$
  FIELDNAME=OFF_CODE    ,ALIAS=OCODE    ,USAGE=A3      ,ACTUAL=A3      ,,$
  FIELDNAME=OFF_STREET  ,ALIAS=OSTR     ,USAGE=A20     ,ACTUAL=A20     ,,$
  FIELDNAME=OFF_CITY    ,ALIAS=OCI      ,USAGE=A15     ,ACTUAL=A15     ,,$
  FIELDNAME=OFF_STATE   ,ALIAS=OST      ,USAGE=A2      ,ACTUAL=A2      ,,$
  GROUP=OFF_FULL_ZIP    ,ALIAS=OFZIP    ,USAGE=A9      ,ACTUAL=A9      ,,$
    FIELDNAME=OFF_ZIP    ,ALIAS=OZIP     ,USAGE=A5      ,ACTUAL=A5      ,,$
    FIELDNAME=OFF_ZIP_L  ,ALIAS=OZIPL    ,USAGE=A4      ,ACTUAL=A4      ,,$
  FIELDNAME=O_PHONES    ,ALIAS=OPOSIT   ,USAGE=A21     ,ACTUAL=A21     ,,$
  FIELDNAME=OFF_AREA_CDE,ALIAS=OAREACD  ,USAGE=A3      ,ACTUAL=A3      ,,$
  FIELDNAME=SPEED_DIAL  ,ALIAS=OSPDL    ,USAGE=A3      ,ACTUAL=A3      ,,$

SEGNAME=PHONES,PARENT=OFFICE,SEGTYPE=S,OCCURS=3,POSITION=O_PHONES,$
  FIELDNAME=OFF_PHONE    ,ALIAS=OPHONE   ,USAGE=A7      ,ACTUAL=A7      ,,$
  FIELDNAME=LINE_NO      ,ALIAS=ORDER    ,USAGE=I4      ,ACTUAL=I4      ,,$

SEGNAME=STRUCTUR,PARENT=EMPLOYE,SEGTYPE=S,$
  FIELDNAME=STRUCTURE_CD,ALIAS=SCDE    ,USAGE=A2      ,ACTUAL=A2      ,,$
  FIELDNAME=STRUCTURE_DT,ALIAS=SDTE    ,USAGE=A6YMD   ,ACTUAL=A6      ,,$

```

```

SEGNAME=SUBORDS , PARENT=STRUCTUR , SEGTYPE=U , $
  FIELDNAME=SUB_ID      , ALIAS=SUBID    , USAGE=A4      , ACTUAL=A4      , $
  GROUP=SUB_NAME        , ALIAS=SUBNAME  , USAGE=A25     , ACTUAL=A25     , $
    FIELDNAME=SUB_F_NAME, ALIAS=SUBFN    , USAGE=A10     , ACTUAL=A10     , $
    FIELDNAME=SUB_L_NAME, ALIAS=SUBLN    , USAGE=A15     , ACTUAL=A15     , $
  FIELDNAME=SUB_STREET  , ALIAS=SUBSTR   , USAGE=A20     , ACTUAL=A20     , $
  FIELDNAME=SUB_CITY    , ALIAS=SUBCI    , USAGE=A15     , ACTUAL=A15     , $
  FIELDNAME=SUB_STATE   , ALIAS=SUBST    , USAGE=A2      , ACTUAL=A2      , $
  GROUP=SUB_FULL_ZIP    , ALIAS=SUBFZIP  , USAGE=A9      , ACTUAL=A9      , $
    FIELDNAME=SUB_ZIP    , ALIAS=SUBZIP   , USAGE=A5      , ACTUAL=A5      , $
    FIELDNAME=SUB_ZIP_L  , ALIAS=SUBZIPL  , USAGE=A4      , ACTUAL=A4      , $
  FIELDNAME=SUB_PHONE   , ALIAS=SUBPHON  , USAGE=A10     , ACTUAL=A10     , $
  FIELDNAME=SUB_STATUS  , ALIAS=SUBSTAT  , USAGE=A2      , ACTUAL=A2      , $
  FIELDNAME=SUB_SSN     , ALIAS=SUBSSN   , USAGE=A9      , ACTUAL=A9      , $
  FIELDNAME=SUB_STRT_DTE, ALIAS=SUBSDTE  , USAGE=A6YMD   , ACTUAL=A6      , $
  FIELDNAME=SUB_TERM_DTE, ALIAS=SUBTDTE  , USAGE=A6YMD   , ACTUAL=A6      , $
  FIELDNAME=SUB_BRTH_DTE, ALIAS=SUBBDTE  , USAGE=A6YMD   , ACTUAL=A6      , $

SEGNAME=EMPOSIT , PARENT=EMPLOYE , SEGTYPE=S , $
  FIELDNAME=POS_STRT_DTE, ALIAS=PSDTE    , USAGE=A6YMD   , ACTUAL=A6      , $
  FIELDNAME=POS_FIN_DTE , ALIAS=PFDTTE   , USAGE=A6YMD   , ACTUAL=A6      , $
  FIELDNAME=SALARY_GRADE, ALIAS=SGRADE   , USAGE=P4      , ACTUAL=Z2      , $
  FIELDNAME=SALARY_AMT  , ALIAS=SAMOUNT  , USAGE=P10.2   , ACTUAL=P5      , $
  FIELDNAME=BONUS_PCT   , ALIAS=BPCT     , USAGE=P4      , ACTUAL=P2      , $
  FIELDNAME=COMMIS_PCT  , ALIAS=CPCT     , USAGE=P4      , ACTUAL=P2      , $
  FIELDNAME=OVERTIME_PCT, ALIAS=ORATE     , USAGE=P5.2    , ACTUAL=P2      , $

SEGNAME=JOB , PARENT=EMPOSIT , SEGTYPE=U , $
  FIELDNAME=JOB_ID      , ALIAS=JID      , USAGE=A4      , ACTUAL=A4      , $
  FIELDNAME=TITLE       , ALIAS=JTIT     , USAGE=A20     , ACTUAL=A20     ,
  FIELDTYPE=I           ,                ,                ,                , $
  DEFINE SHORTTITLE/A10 = EDIT(JTIT, '999999999$') ; , $
  FIELDNAME=JOB_DESC    , ALIAS=JDESC    , USAGE=A120    , ACTUAL=A120    , $
  FIELDNAME=REQUIREMENTS, ALIAS=JREQ     , USAGE=A120    , ACTUAL=A120    , $
  FIELDNAME=MIN_SALARY  , ALIAS=MINSAL   , USAGE=P12.2   , ACTUAL=Z8      , $
  FIELDNAME=MAX_SALARY  , ALIAS=MAXSAL   , USAGE=P12.2   , ACTUAL=Z8      , $
  FIELDNAME=SAL_GRADE_1 , ALIAS=GRADE1   , USAGE=P4      , ACTUAL=Z2      , $
  FIELDNAME=SAL_GRADE_2 , ALIAS=GRADE1   , USAGE=P4      , ACTUAL=Z2      , $
  FIELDNAME=SAL_GRADE_3 , ALIAS=GRADE1   , USAGE=P4      , ACTUAL=Z2      , $
  FIELDNAME=SAL_GRADE_4 , ALIAS=GRADE1   , USAGE=P4      , ACTUAL=Z2      , $
  FIELDNAME=POSITION_NUM, ALIAS=NOPOS    , USAGE=P4      , ACTUAL=Z3      , $
  FIELDNAME=NUM_OPEN    , ALIAS=NOOPEN   , USAGE=P4      , ACTUAL=Z3      , $

SEGNAME=EXPERTSE , PARENT=EMPLOYE , SEGTYPE=S , $
  FIELDNAME=SKILL_LEVEL , ALIAS=SKLVL    , USAGE=A2      , ACTUAL=A2      , $
  FIELDNAME=EXPERT_DTE  , ALIAS=EXDTE    , USAGE=A6YMD   , ACTUAL=A6      , $

SEGNAME=SKILL , PARENT=EXPERTSE , SEGTYPE=U , $
  FIELDNAME=SKILL_ID    , ALIAS=SKID     , USAGE=A4      , ACTUAL=A4      , $
  FIELDNAME=SKILL_NAME  , ALIAS=SKNAME   , USAGE=A12     , ACTUAL=A12     ,
  FIELDTYPE=I           ,                ,                ,                , $
  FIELDNAME=SKILL_DESC  , ALIAS=SKDESC   , USAGE=A60     , ACTUAL=A60     , $

```



```

SEGNAME=COVERAGE, PARENT=EMPLOYE, SEGTYPE=S, $
  FIELDNAME=COV_SEL_DT, ALIAS=SELDTE, USAGE=I6YMD, ACTUAL=Z6, $
  FIELDNAME=COV_TERM_DTE, ALIAS=TERMDTE, USAGE=A6YMD, ACTUAL=A6, $
  FIELDNAME=COVER_TYPE, ALIAS=COVTYP, USAGE=A1, ACTUAL=A1, $
  FIELDNAME=COV_CODE, ALIAS=CPCODE, USAGE=A3, ACTUAL=A3, $

SEGNAME=HOSPITAL, PARENT=COVERAGE, SEGTYPE=S, $
  FIELDNAME=H_CLAIM_DTE, ALIAS=HCDTE, USAGE=I6YMD, ACTUAL=Z6, $
  FIELDNAME=H_FIRST_NAME, ALIAS=HFN, USAGE=A10, ACTUAL=A10, $
  FIELDNAME=H_LAST_NAME, ALIAS=DLN, USAGE=A15, ACTUAL=A15, $
  FIELDNAME=H_BIRTH_DTE, ALIAS=DBDTE, USAGE=I6YMD, ACTUAL=Z6, $
  FIELDNAME=H_SEX, ALIAS=DSEX, USAGE=A1, ACTUAL=A1, $
  FIELDNAME=H_RELATED_BY, ALIAS=DREL, USAGE=A10, ACTUAL=A10, $
  FIELDNAME=HOSP_NAME, ALIAS=HOSPNM, USAGE=A25, ACTUAL=A25, $
  FIELDNAME=HOSP_STREET, ALIAS=HSTR, USAGE=A20, ACTUAL=A20, $
  FIELDNAME=HOSP_CITY, ALIAS=HCITY, USAGE=A15, ACTUAL=A15, $
  FIELDNAME=HOSP_STATE, ALIAS=HSTATE, USAGE=A2, ACTUAL=A2, $
  GROUP=HOSP_FUL_ZIP, ALIAS=HFZIP, USAGE=A9, ACTUAL=A9, $
    FIELDNAME=HOSP_ZIP, ALIAS=HZIP, USAGE=A5, ACTUAL=A5, $
    FIELDNAME=HOSP_ZIP_L, ALIAS=HZIPL, USAGE=A4, ACTUAL=A4, $
  FIELDNAME=ADMITTED, ALIAS=HADTE, USAGE=I6YMD, ACTUAL=Z6, $
  FIELDNAME=DISCHARGED, ALIAS=HDDTE, USAGE=I6YMD, ACTUAL=Z6, $
  FIELDNAME=H_DIAGNOSIS1, ALIAS=HDIAG1, USAGE=A60, ACTUAL=A60, $
  FIELDNAME=H_DIAGNOSIS2, ALIAS=HDIAG2, USAGE=A60, ACTUAL=A60, $
  FIELDNAME=WARD_DAYS, ALIAS=WDAYS, USAGE=P5, ACTUAL=P3, $
  FIELDNAME=WARD_RATE, ALIAS=WRATE, USAGE=P10.2, ACTUAL=P5, $
  FIELDNAME=WARD_TOTAL, ALIAS=WTOT, USAGE=P10.2, ACTUAL=P5, $
  FIELDNAME=SEMI_DAYS, ALIAS=SDAYS, USAGE=P5, ACTUAL=P3, $
  FIELDNAME=SEMI_RATE, ALIAS=SRATE, USAGE=P10.2, ACTUAL=P5, $
  FIELDNAME=SEMI_TOTAL, ALIAS=STOT, USAGE=P10.2, ACTUAL=P5, $
  FIELDNAME=DELIVERY_TOT, ALIAS=DELCST, USAGE=P10.2, ACTUAL=P5, $
  FIELDNAME=ANESTHES_TOT, ALIAS=ANESTH, USAGE=P10.2, ACTUAL=P5, $
  FIELDNAME=LAB_TOT, ALIAS=LABCOST, USAGE=P10.2, ACTUAL=P5, $
  FIELDNAME=, ALIAS=, USAGE=A4, ACTUAL=A4, $
  FIELDNAME=CLAIM_MONTH, ALIAS=CMO, USAGE=I2, ACTUAL=Z2, $

```

```

SEGNAME=NON_HOSP, SEGTYPE=S, PARENT=COVERAGE, $
  FIELDNAME=N_CLAIM_DTE , ALIAS=NCDTE , USAGE=I6YMD, ACTUAL=Z6 , $
  FIELDNAME=N_FIRST_NAME, ALIAS=NFN , USAGE=A10 , ACTUAL=A10 , $
  FIELDNAME=N_LAST_NAME , ALIAS=NLN , USAGE=A15 , ACTUAL=A15 , $
  FIELDNAME=N_BIRTH_DTE , ALIAS=NBDTE , USAGE=I6YMD, ACTUAL=Z6 , $
  FIELDNAME=N_SEX , ALIAS=NSEX , USAGE=A1 , ACTUAL=A1 , $
  FIELDNAME=N_RELATED_BY, ALIAS=NREL , USAGE=A10 , ACTUAL=A10 , $
  FIELDNAME=PHYS_FNAME , ALIAS=PHYSFN , USAGE=A10 , ACTUAL=A10 , $
  FIELDNAME=PHYS_LNAME , ALIAS=PHYSLN , USAGE=A15 , ACTUAL=A15 , $
  FIELDNAME=PHYS_STREET , ALIAS=PSTREET , USAGE=A20 , ACTUAL=A20 , $
  FIELDNAME=PHYS_CITY , ALIAS=PCITY , USAGE=A15 , ACTUAL=A15 , $
  FIELDNAME=PHYS_STATE , ALIAS=PSTATE , USAGE=A2 , ACTUAL=A2 , $
  GROUP=PHYS_FUL_ZIP , ALIAS=PFZIP , USAGE=A9 , ACTUAL=A9 , $
    FIELDNAME=PHYS_ZIP , ALIAS=PZIP , USAGE=A5 , ACTUAL=A5 , $
    FIELDNAME=PHYS_ZIP_L, ALIAS=PZIPL , USAGE=A4 , ACTUAL=A4 , $
  FIELDNAME=PHYS_ID , ALIAS=PHYSID , USAGE=P6 , ACTUAL=Z6 , $
  FIELDNAME=P_DIAGNOSIS1, ALIAS=PDIAG1 , USAGE=A60 , ACTUAL=A60 , $
  FIELDNAME=P_DIAGNOSIS2, ALIAS=PDIAG2 , USAGE=A60 , ACTUAL=A60 , $
  FIELDNAME=P_NO_OF_PROC, ALIAS=DENTNOP , USAGE=I2 , ACTUAL=I2 , $
  FIELDNAME= , ALIAS= , USAGE=A1 , ACTUAL=A1 , $

SEGNAME=PHYSCHRG, SEGTYPE=S, PARENT=NON_HOSP, OCCURS=P_NO_OF_PROC, $
  FIELDNAME=P_SERVICE_DT, ALIAS=PSERVDT , USAGE=I6YMD, ACTUAL=Z6 , $
  FIELDNAME=PHYS_PROC_CD, ALIAS=PCODE , USAGE=P4 , ACTUAL=Z4 , $
  FIELDNAME=P_SERV_DESC , ALIAS=PSRVDSC , USAGE=A60 , ACTUAL=A60 , $
  FIELDNAME=PHYS_FEE , ALIAS=PFEE , USAGE=P11.2 , ACTUAL=P5 , $
  FIELDNAME= , ALIAS= , USAGE=A1 , ACTUAL=A1 , $
  FIELDNAME=PHYS_CHRG_NO, ALIAS=ORDER , USAGE=I4 , ACTUAL=I4 , $

SEGNAME=DENTAL, SEGTYPE=S, PARENT=COVERAGE, $
  FIELDNAME=D_CLAIM_DTE , ALIAS=DCDTE , USAGE=I6YMD, ACTUAL=Z6 , $
  FIELDNAME=D_FIRST_NAME, ALIAS=DFN , USAGE=A10 , ACTUAL=A10 , $
  FIELDNAME=D_LAST_NAME , ALIAS=DLN , USAGE=A15 , ACTUAL=A15 , $
  FIELDNAME=D_BIRTH_DTE , ALIAS=DBDTE , USAGE=I6YMD, ACTUAL=Z6 , $
  FIELDNAME=D_SEX , ALIAS=DSEX , USAGE=A1 , ACTUAL=A1 , $
  FIELDNAME=D_RELATED_BY, ALIAS=DREL , USAGE=A10 , ACTUAL=A10 , $
  FIELDNAME=DENT_FNAME , ALIAS=DENTFN , USAGE=A10 , ACTUAL=A10 , $
  FIELDNAME=DENT_LNAME , ALIAS=DENTLN , USAGE=A15 , ACTUAL=A15 , $
  FIELDNAME=DENT_STREET , ALIAS=DSTREET , USAGE=A20 , ACTUAL=A20 , $
  FIELDNAME=DENT_CITY , ALIAS=DCITY , USAGE=A15 , ACTUAL=A15 , $
  FIELDNAME=DENT_STATE , ALIAS=DSTATE , USAGE=A2 , ACTUAL=A2 , $
  GROUP=DENT_FUL_ZIP , ALIAS=DFZIP , USAGE=A9 , ACTUAL=A9 , $
    FIELDNAME=DENT_ZIP , ALIAS=DZIP , USAGE=A5 , ACTUAL=A5 , $
    FIELDNAME=DENT_ZIP_L, ALIAS=DZIPL , USAGE=A4 , ACTUAL=A4 , $
  FIELDNAME=DENT_LICENSE, ALIAS=DENTLI , USAGE=P6 , ACTUAL=Z6 , $
  FIELDNAME=D_NO_OF_PROC, ALIAS=DENTNOP , USAGE=I2 , ACTUAL=I2 , $
  FIELDNAME= , ALIAS= , USAGE=A3 , ACTUAL=A3 , $

```

```

SEGNAME=DENTCHRG, SEGTYPE=S, PARENT=DENTAL, OCCURS=D_NO_OF_PROC, $
  FIELDNAME=TOOTH_NUM, ALIAS=TNUM, USAGE=P2, ACTUAL=Z2, $
  FIELDNAME=D_SERVICE_DT, ALIAS=DSERVDT, USAGE=A6YMD, ACTUAL=A6, $
  FIELDNAME=DENT_PROC_CD, ALIAS=DCODE, USAGE=P4, ACTUAL=Z4, $
  FIELDNAME=D_SERV_DESC, ALIAS=DSRVDSC, USAGE=A60, ACTUAL=A60, $
  FIELDNAME=DENT_FEE, ALIAS=DFEE, USAGE=P11.2, ACTUAL=P5, $
  FIELDNAME=, ALIAS=, USAGE=A3, ACTUAL=A3, $
  FIELDNAME=DENT_CHRG_NO, ALIAS=ORDER, USAGE=I9, ACTUAL=I4, $

SEGNAME=INSURNC, PARENT=COVERAGE, SEGTYPE=U, $
  FIELDNAME=INS_PLAN_CDE, ALIAS=ICODE, USAGE=A3, ACTUAL=A3, $
  FIELDNAME=INS_CO_NAME, ALIAS=ICONAME, USAGE=A45, ACTUAL=A45, $
  FIELDNAME=INS_STREET, ALIAS=ISTR, USAGE=A20, ACTUAL=A20, $
  FIELDNAME=INS_CITY, ALIAS=ICI, USAGE=A15, ACTUAL=A15, $
  FIELDNAME=INS_STATE, ALIAS=IST, USAGE=A2, ACTUAL=A2, $
  GROUP=INS_FULL_ZIP, ALIAS=IFZIP, USAGE=A9, ACTUAL=A9, $
    FIELDNAME=INS_ZIP, ALIAS=IZIP, USAGE=A5, ACTUAL=A5, $
    FIELDNAME=INS_ZIP_L, ALIAS=IZIPL, USAGE=A4, ACTUAL=A4, $
  FIELDNAME=INS_PHONE, ALIAS=IPHON, USAGE=A10, ACTUAL=A10, $
  FIELDNAME=INS_GROUP_NO, ALIAS=IGROUP, USAGE=A6, ACTUAL=A6, $
  FIELDNAME=DEDUCT, ALIAS=IDED, USAGE=P12.2, ACTUAL=P5, $
  FIELDNAME=MAX_LIFE_CST, ALIAS=MAXCOST, USAGE=P12.2, ACTUAL=P5, $
  FIELDNAME=FAMILY_COST, ALIAS=FCOST, USAGE=P12.2, ACTUAL=P5, $
  FIELDNAME=DEPENDNT_CST, ALIAS=DEPCOST, USAGE=P12.2, ACTUAL=P5, $

```

Access File for Network Sample

This Access File is associated with network subschema EMPSS01 and corresponds to its Master File:

```
SSSCHEMA=EMPSS01,RELEASE=14.0,MODE=DML,TRACE=NO,READY=,$
SEGNAM=DEPT,RECORD=DEPARTMENT,AREA=ORG-DEMO-REGION,
  CLCFLD=DEPT_ID,CLCDUP=N,$
SEGNAM=EMPLOYEE,RECORD=EMPLOYEE,AREA=EMP-DEMO-REGION,
  CLCFLD=EMP_ID,CLCDUP=N,ACCESS=SET,SETNAME=DEPT-EMPLOYEE,
  SETMBR=OA,GETOWN=Y,MULTMBR=N,$
SEGNAM=OFFICE,RECORD=OFFICE,AREA=ORG-DEMO-REGION,
  CLCFLD=OFF_CODE,CLCDUP=N,ACCESS=SET,SETNAME=OFFICE-EMPLOYEE,
  SETMBR=OA,GETOWN=Y,MULTMBR=N,$
SEGNAM=STRUCTUR,RECORD=STRUCTURE,AREA=EMP-DEMO-REGION,
  ACCESS=SET,SETNAME=MANAGES,SETMBR=MA,GETOWN=Y,MULTMBR=N,$
SEGNAM=SUBORDS,RECORD=EMPLOYEE,AREA=EMP-DEMO-REGION,
  CLCFLD=SUB_ID,CLCDUP=N,ACCESS=SET,SETNAME=REPORTS-TO,
  SETMBR=OM,GETOWN=Y,MULTMBR=N,$
SEGNAM=EMPOSIT,RECORD=EMPOSITION,AREA=EMP-DEMO-REGION,
  ACCESS=SET,SETNAME=EMP-EMPOSITION,SETMBR=MA,GETOWN=Y,MULTMBR=N,$
SEGNAM=JOB,RECORD=JOB,AREA=ORG-DEMO-REGION,
  CLCFLD=JOB_ID,CLCDUP=N,ACCESS=SET,SETNAME=JOB-EMPOSITION,
  SETMBR=OM,GETOWN=Y,MULTMBR=N,SEQFIELD=TITLE,$
SEGNAM=EXPERTSE,RECORD=EXPERTISE,AREA=EMP-DEMO-REGION,
  ACCESS=SET,SETNAME=EMP-EXPERTISE,KEYFLD=SKILL_LEVEL,SETORD=D,
  SETDUP=Y,SETMBR=MA,GETOWN=Y,MULTMBR=N,$
SEGNAM=SKILL,RECORD=SKILL,AREA=ORG-DEMO-REGION,
  CLCFLD=SKILL_ID,CLCDUP=N,ACCESS=SET,SETNAME=SKILL-EXPERTISE,
  KEYFLD=SKILL_LEVEL,SETORD=D,SETDUP=Y,
  SETMBR=MA,GETOWN=Y,MULTMBR=N,SEQFIELD=SKILL_NAME,$
SEGNAM=COVERAGE,RECORD=COVERAGE,AREA=INS-DEMO-REGION,
  ACCESS=SET,SETNAME=EMP-COVERAGE,SETMBR=MA,GETOWN=Y,MULTMBR=N,$
SEGNAM=HOSPITAL,RECORD=HOSPITAL-CLAIM,AREA=INS-DEMO-REGION,
  ACCESS=SET,SETNAME=COVERAGE-CLAIMS,SETMBR=MA,GETOWN=Y,MULTMBR=Y,$
SEGNAM=NON_HOSP,RECORD=NON-HOSP-CLAIM,AREA=INS-DEMO-REGION,
  ACCESS=SET,SETNAME=COVERAGE-CLAIMS,SETMBR=MA,GETOWN=Y,MULTMBR=Y,$
SEGNAM=DENTAL,RECORD=DENTAL-CLAIM,AREA=INS-DEMO-REGION,
  ACCESS=SET,SETNAME=COVERAGE-CLAIMS,SETMBR=MA,GETOWN=Y,MULTMBR=Y,$
IXSET=JOB-TITLE-NDX,IXFLD=TITLE,IXDUP=N,IXORD=A,
  IXAREA=INS-DEMO-REGION,$
IXSET=SKILL-NAME-NDX,IXFLD=SKILL_NAME,IXDUP=N,IXORD=D,
  IXAREA=EMP-DEMO-REGION,$
SEGNAM=INSURNCE,RECORD=INSURANCE-PLAN,AREA=INS-DEMO-REGION,
  CLCFLD=INS_PLAN_CD
E,CLCDUP=N,ACCESS=CLC,KEYFLD=COV_CODE,$
```

Master File for LRF Sample

This Master File corresponds to LRF subschema EMPSS02:

```

FILE=EMPDATA, SUFFIX=IDMSR, $
SEGNAME=DEPTEMPO, $
    FIELD=DEPT_ID, ALIAS=DID, USAGE=A4, ACTUAL=A4, $
    FIELD=DEPT_NAME, ALIAS=DNAME, USAGE=A45, ACTUAL=A45, $
    FIELD=DEPT_HEAD, ALIAS=DHEAD, USAGE=A4, ACTUAL=A4, $
    FIELD=, ALIAS=FILL.END, USAGE=A3, ACTUAL=A3, $
    FIELD=EMP_ID, ALIAS=EID, USAGE=A4, ACTUAL=A4, $
    GROUP=EMP_NAME, ALIAS=ENAME, USAGE=A25, ACTUAL=A25, $
        FIELD=FIRST_NAME, ALIAS=EFN, USAGE=A10, ACTUAL=A10, $
        FIELD=LAST_NAME, ALIAS=ELN, USAGE=A15, ACTUAL=A15, $
    FIELD=EMP_STREET, ALIAS=ESTR, USAGE=A20, ACTUAL=A20, $
    FIELD=EMP_CITY, ALIAS=ECI, USAGE=A15, ACTUAL=A15, $
    FIELD=EMP_STATE, ALIAS=EST, USAGE=A2, ACTUAL=A2, $
    GROUP=EMP_FULL_ZIP, ALIAS=EFZIP, USAGE=A9, ACTUAL=A9, $
        FIELD=EMP_ZIP, ALIAS=EZIP, USAGE=A5, ACTUAL=A5, $
        FIELD=EMP_ZIP_L, ALIAS=EZIPL, USAGE=A4, ACTUAL=A4, $
    FIELD=EMP_PHONE, ALIAS=EPHON, USAGE=A10, ACTUAL=A10, $
    FIELD=STATUS, ALIAS=ESTAT, USAGE=A2, ACTUAL=A2, $
    FIELD=SOC_SEC_NUM, ALIAS=SSN, USAGE=A9, ACTUAL=A9, $
    FIELD=EMP_STRT_DTE, ALIAS=ESDTE, USAGE=A6YMD, ACTUAL=A6, $
    FIELD=EMP_TERM_DTE, ALIAS=ETDTE, USAGE=A6YMD, ACTUAL=A6, $
    FIELD=EMP_BRTH_DTE, ALIAS=EBDTE, USAGE=A6YMD, ACTUAL=A6, $
    FIELD=, ALIAS=FILL.END, USAGE=A6, ACTUAL=A6, $
    FIELD=POS_STRT_DT1, ALIAS=PSDTE, USAGE=A6YMD, ACTUAL=A6, $
    FIELD=POS_FIN_DT1, ALIAS=PFDT, USAGE=A6YMD, ACTUAL=A6, $
    FIELD=SALARY_GRAD1, ALIAS=SGRADE, USAGE=P4, ACTUAL=Z2, $
    FIELD=SALARY_AMT1, ALIAS=SAMOUNT, USAGE=P10.2, ACTUAL=P5, $
    FIELD=BONUS_PCT1, ALIAS=BPCT, USAGE=P4, ACTUAL=P2, $
    FIELD=COMMIS_PCT1, ALIAS=CPCT, USAGE=P4, ACTUAL=P2, $
    FIELD=OVERTIM_PCT1, ALIAS=ORATE, USAGE=P5.2, ACTUAL=P2, $
SEGNAME=JOBPOS, PARENT=DEPTEMPO, SEGTYPE=U, $
    FIELD=JOB_ID, ALIAS=JID, USAGE=A4, ACTUAL=A4, $
    FIELD=TITLE, ALIAS=JTIT, USAGE=A20, ACTUAL=A20, $
    FIELD=JOB_DESC, ALIAS=JDESC, USAGE=A120, ACTUAL=A120, $
    FIELD=REQUIREMENTS, ALIAS=JREQ, USAGE=A120, ACTUAL=A120, $
    FIELD=MIN_SALARY, ALIAS=MINSAL, USAGE=P12.2, ACTUAL=Z8, $
    FIELD=MAX_SALARY, ALIAS=MAXSAL, USAGE=P12.2, ACTUAL=Z8, $
    FIELD=SAL_GRADE_1, ALIAS=GRADE1, USAGE=P4, ACTUAL=Z2, $
    FIELD=SAL_GRADE_2, ALIAS=GRADE1, USAGE=P4, ACTUAL=Z2, $
    FIELD=SAL_GRADE_3, ALIAS=GRADE1, USAGE=P4, ACTUAL=Z2, $
    FIELD=SAL_GRADE_4, ALIAS=GRADE1, USAGE=P4, ACTUAL=Z2, $
    FIELD=POSITION_NUM, ALIAS=NOPOS, USAGE=P4, ACTUAL=Z3, $
    FIELD=NUM_OPEN, ALIAS=NOOPEN, USAGE=P4, ACTUAL=Z3, $
    FIELD=, ALIAS=FILL.END, USAGE=A2, ACTUAL=A2, $
    FIELD=POS_STRT_DT2, ALIAS=PSDTE, USAGE=A6YMD, ACTUAL=A6, $
    FIELD=POS_FIN_DT2, ALIAS=PFDT, USAGE=A6YMD, ACTUAL=A6, $
    FIELD=SALARY_GRAD2, ALIAS=SGRADE, USAGE=P4, ACTUAL=Z2, $
    FIELD=SALARY_AMT2, ALIAS=SAMOUNT, USAGE=P10.2, ACTUAL=P5, $
    FIELD=BONUS_PCT2, ALIAS=BPCT, USAGE=P4, ACTUAL=P2, $
    FIELD=COMMIS_PCT2, ALIAS=CPCT, USAGE=P4, ACTUAL=P2, $
    FIELD=OVERTIM_PCT2, ALIAS=ORATE, USAGE=P5.2, ACTUAL=P2, $

```

Access File for LRF Sample

This Access File is associated with LRF subschema EMPSS02 and corresponds to its Master File:

```
SSHEMA=EMPSS02,RELEASE=14,MODE=LR,TRACE=PARMS,READY=ALL,$  
SEGNAM=DEPTempo,RECORD=DEPT-EMP-POS,AREA=EMP-DEMO-REGION,  
  LR=Y,$  
SEGNAM=JOBPOS,RECORD=JOB-EMPOSITION,AREA=ORG-DEMO-REGION,  
  LR=Y,  
ACCESS=LR,KEYFLD=POS_STRT_DT1,IXFLD=POS_STRT_DT2,$
```

Sample of a Partial LRF Record

The following is an example of a NULL SELECT clause that creates a partial record by returning a user-defined record code. The data adapter does not support this user-defined code or any status code other than LR-FOUND or LR-NOT-FOUND.

```
SELECT  
  OBTAIN EACH JOB WITHIN ORG-DEMO-REGION  
  IF JOB-EMPOSITION IS NOT EMPTY  
    ON 0000 RETURN NO-POS-FOR-JOB  
  OBTAIN EACH EMPOSITION WITHIN JOB-EMPOSITION.
```

EMPFULL Master File

This Master File is used in the examples throughout this manual. For instructions about creating this Master File, see the sample session in Chapter 6, *Creating File Descriptions With AUTOIDMS*:

```
$$$ CREATED BY AUTOIDMS ON 02/13/01 AT 14.32.16 BY PMSSAE
$$$ SSHEMA=EMPSS01
FILE=EMPFULL,SUFFIX=IDMSR,$
```

```
SEGNAME=S0410_01,SEGTYPE=S,$
  FIELD=DEPT_ID                ,ALIAS=F0410_1    ,A4    ,A4    ,,$
  FIELD=DEPT_NAME              ,ALIAS=F0410_2    ,A45    ,A45    ,,$
  FIELD=DEPT_HEAD_ID           ,ALIAS=F0410_3    ,A4    ,A4    ,,$
  FIELD=FIL 0003               ,ALIAS=F0410_4    ,A3    ,A3    ,,$
  FIELD=S0410_01_KEY           ,ALIAS=DBKEY      ,I10    ,I4    ,,$
```

```
SEGNAME=S0415_02,SEGTYPE=S,PARENT=S0410_01,$
  FIELD=EMP_ID                ,ALIAS=F0415_1    ,A4    ,A4    ,,$
  GROUP=EMP_NAME              ,ALIAS=F0415_2    ,A25    ,A25    ,,$
  FIELD=EMP_FIRST_NAME        ,ALIAS=F0415_3    ,A10    ,A10    ,,$
  FIELD=EMP_LAST_NAME         ,ALIAS=F0415_4    ,A15    ,A15    ,,$
  GROUP=EMP_ADDRESS           ,ALIAS=F0415_5    ,A46    ,A46    ,,$
  FIELD=EMP_STREET            ,ALIAS=F0415_6    ,A20    ,A20    ,,$
  FIELD=EMP_CITY              ,ALIAS=F0415_7    ,A15    ,A15    ,,$
  FIELD=EMP_STATE             ,ALIAS=F0415_8    ,A2    ,A2    ,,$
  GROUP=EMP_ZIP               ,ALIAS=F0415_9    ,A9    ,A9    ,,$
  FIELD=EMP_ZIP_FIRST_FIVE    ,ALIAS=F0415_10   ,A5    ,A5    ,,$
  FIELD=EMP_ZIP_LAST_FOUR     ,ALIAS=F0415_11   ,A4    ,A4    ,,$
  FIELD=EMP_PHONE             ,ALIAS=F0415_12   ,A10    ,A10    ,,$
  FIELD=STATUS                ,ALIAS=F0415_13   ,A2    ,A2    ,,$
  FIELD=SS_NUMBER             ,ALIAS=F0415_19   ,A9    ,A9    ,,$
  GROUP=START_DATE           ,ALIAS=F0415_20   ,A8    ,A8    ,,$
  FIELD=START_YEAR            ,ALIAS=F0415_21   ,A4    ,A4    ,,$
  FIELD=START_MONTH           ,ALIAS=F0415_22   ,A2    ,A2    ,,$
  FIELD=START_DAY             ,ALIAS=F0415_23   ,A2    ,A2    ,,$
  GROUP=TERMINATION_DATE     ,ALIAS=F0415_24   ,A8    ,A8    ,,$
  FIELD=TERMINATION_YEAR     ,ALIAS=F0415_25   ,A4    ,A4    ,,$
  FIELD=TERMINATION_MONTH    ,ALIAS=F0415_26   ,A2    ,A2    ,,$
  FIELD=TERMINATION_DAY      ,ALIAS=F0415_27   ,A2    ,A2    ,,$
  GROUP=BIRTH_DATE           ,ALIAS=F0415_28   ,A8    ,A8    ,,$
  FIELD=BIRTH_YEAR            ,ALIAS=F0415_29   ,A4    ,A4    ,,$
  FIELD=BIRTH_MONTH           ,ALIAS=F0415_30   ,A2    ,A2    ,,$
  FIELD=BIRTH_DAY             ,ALIAS=F0415_31   ,A2    ,A2    ,,$
  FIELD=S0415_02_KEY         ,ALIAS=DBKEY      ,I10    ,I4    ,,$
```

```

SEGNAME=S0400_03,SEGTYPE=S,PARENT=S0415_02,$
GROUP=SELECTION_DATE                ,ALIAS=F0400_1      ,A8      ,A8      ,$
FIELD=SELECTION_YEAR                ,ALIAS=F0400_2      ,A4      ,A4      ,$
FIELD=SELECTION_MONTH               ,ALIAS=F0400_3      ,A2      ,A2      ,$
FIELD=SELECTION_DAY                 ,ALIAS=F0400_4      ,A2      ,A2      ,$
GROUP=TERMINATION_DATE              ,ALIAS=F0400_5      ,A8      ,A8      ,$
FIELD=TERMINATION_YEAR              ,ALIAS=F0400_6      ,A4      ,A4      ,$
FIELD=TERMINATION_MONTH             ,ALIAS=F0400_7      ,A2      ,A2      ,$
FIELD=TERMINATION_DAY               ,ALIAS=F0400_8      ,A2      ,A2      ,$
FIELD=TYPE                          ,ALIAS=F0400_9      ,A1      ,A1      ,$
FIELD=INS_PLAN_CODE                 ,ALIAS=F0400_13     ,A3      ,A3      ,$
FIELD=S0400_03_KEY                  ,ALIAS=DBKEY        ,I10     ,I4      ,$

SEGNAME=S0420_04,SEGTYPE=S,PARENT=S0415_02,$
GROUP=START_DATE                    ,ALIAS=F0420_1      ,A8      ,A8      ,$
FIELD=START_YEAR                    ,ALIAS=F0420_2      ,A4      ,A4      ,$
FIELD=START_MONTH                   ,ALIAS=F0420_3      ,A2      ,A2      ,$
FIELD=START_DAY                     ,ALIAS=F0420_4      ,A2      ,A2      ,$
GROUP=FINISH_DATE                   ,ALIAS=F0420_5      ,A8      ,A8      ,$
FIELD=FINISH_YEAR                   ,ALIAS=F0420_6      ,A4      ,A4      ,$
FIELD=FINISH_MONTH                  ,ALIAS=F0420_7      ,A2      ,A2      ,$
FIELD=FINISH_DAY                    ,ALIAS=F0420_8      ,A2      ,A2      ,$
FIELD=SALARY_GRADE                  ,ALIAS=F0420_9      ,A2      ,A2      ,$
FIELD=SALARY_AMOUNT                 ,ALIAS=F0420_10     ,P11.2   ,P5      ,$
FIELD=BONUS_PERCENT                 ,ALIAS=F0420_11     ,P5.3     ,P2      ,$
FIELD=COMMISSION_PERCENT            ,ALIAS=F0420_12     ,P5.3     ,P2      ,$
FIELD=OVERTIME_RATE                 ,ALIAS=F0420_13     ,P5.2     ,P2      ,$
FIELD=FIL 0003                      ,ALIAS=F0420_14     ,A3      ,A3      ,$
FIELD=S0420_04_KEY                  ,ALIAS=DBKEY        ,I10     ,I4      ,$

SEGNAME=S0425_05,SEGTYPE=S,PARENT=S0415_02,$
FIELD=SKILL_LEVEL                   ,ALIAS=F0425_1      ,A2      ,A2      ,$
GROUP=EXPERTISE_DATE                ,ALIAS=F0425_6      ,A8      ,A8      ,$
FIELD=EXPERTISE_YEAR                ,ALIAS=F0425_7      ,A4      ,A4      ,$
FIELD=EXPERTISE_MONTH               ,ALIAS=F0425_8      ,A2      ,A2      ,$
FIELD=EXPERTISE_DAY                 ,ALIAS=F0425_9      ,A2      ,A2      ,$
FIELD=FIL 0002                      ,ALIAS=F0425_10     ,A2      ,A2      ,$
FIELD=S0425_05_KEY                  ,ALIAS=DBKEY        ,I10     ,I4      ,$

SEGNAME=S0460_06,SEGTYPE=S,PARENT=S0415_02,$
FIELD=STRUCTURE_CODE                ,ALIAS=F0460_1      ,A2      ,A2      ,$
GROUP=STRUCTURE_DATE                ,ALIAS=F0460_4      ,A8      ,A8      ,$
FIELD=STRUCTURE_YEAR                ,ALIAS=F0460_5      ,A4      ,A4      ,$
FIELD=STRUCTURE_MONTH               ,ALIAS=F0460_6      ,A2      ,A2      ,$
FIELD=STRUCTURE_DAY                 ,ALIAS=F0460_7      ,A2      ,A2      ,$
FIELD=FIL 0002                      ,ALIAS=F0460_8      ,A2      ,A2      ,$
FIELD=S0460_06_KEY                  ,ALIAS=DBKEY        ,I10     ,I4      ,$

```



```

SEGNAME=S0450_07,SEGTYPE=U,PARENT=S0415_02,$
FIELD=OFFICE_CODE,ALIAS=F0450_1,A3,A3,$
GROUP=OFFICE_ADDRESS,ALIAS=F0450_2,A46,A46,$
FIELD=OFFICE_STREET,ALIAS=F0450_3,A20,A20,$
FIELD=OFFICE_CITY,ALIAS=F0450_4,A15,A15,$
FIELD=OFFICE_STATE,ALIAS=F0450_5,A2,A2,$
GROUP=OFFICE_ZIP,ALIAS=F0450_6,A9,A9,$
FIELD=OFFICE_ZIP_FIRST_FIVE,ALIAS=F0450_7,A5,A5,$
FIELD=OFFICE_ZIP_LAST_FOUR,ALIAS=F0450_8,A4,A4,$
FIELD=S07_OCC01,ALIAS=,A21,A21,$
FIELD=OFFICE_AREA_CODE,ALIAS=F0450_10,A3,A3,$
FIELD=SPEED_DIAL,ALIAS=F0450_11,A3,A3,$
FIELD=S0450_07_KEY,ALIAS=DBKEY,I10,I4,$

SEGNAME=OCC07_01,SEGTYPE=S,PARENT=S0450_07,
OCCURS=0003,POSITION=S07_OCC01,$
FIELD=OFFICE_PHONE,ALIAS=F0450_9,A7,A7,$
FIELD=OCC07_01,ALIAS=ORDER,I9,I4,$

SEGNAME=S0405_08,SEGTYPE=S,PARENT=S0400_03,$
GROUP=CLAIM_DATE,ALIAS=F0405_1,A8,A8,$
FIELD=CLAIM_YEAR,ALIAS=F0405_2,A4,A4,$
FIELD=CLAIM_MONTH,ALIAS=F0405_3,A2,A2,$
FIELD=CLAIM_DAY,ALIAS=F0405_4,A2,A2,$
GROUP=PATIENT_NAME,ALIAS=F0405_5,A25,A25,$
FIELD=PATIENT_FIRST_NAME,ALIAS=F0405_6,A10,A10,$
FIELD=PATIENT_LAST_NAME,ALIAS=F0405_7,A15,A15,$
GROUP=PATIENT_BIRTH_DATE,ALIAS=F0405_8,A8,A8,$
FIELD=PATIENT_BIRTH_YEAR,ALIAS=F0405_9,A4,A4,$
FIELD=PATIENT_BIRTH_MONTH,ALIAS=F0405_10,A2,A2,$
FIELD=PATIENT_BIRTH_DAY,ALIAS=F0405_11,A2,A2,$
FIELD=PATIENT_SEX,ALIAS=F0405_12,A1,A1,$
FIELD=RELATION_TO_EMPLOYEE,ALIAS=F0405_13,A10,A10,$
GROUP=DENTIST_NAME,ALIAS=F0405_14,A25,A25,$
FIELD=DENTIST_FIRST_NAME,ALIAS=F0405_15,A10,A10,$
FIELD=DENTIST_LAST_NAME,ALIAS=F0405_16,A15,A15,$
GROUP=DENTIST_ADDRESS,ALIAS=F0405_17,A46,A46,$
FIELD=DENTIST_STREET,ALIAS=F0405_18,A20,A20,$
FIELD=DENTIST_CITY,ALIAS=F0405_19,A15,A15,$
FIELD=DENTIST_STATE,ALIAS=F0405_20,A2,A2,$
GROUP=DENTIST_ZIP,ALIAS=F0405_21,A9,A9,$
FIELD=DENTIST_ZIP_FIRST_FIVE,ALIAS=F0405_22,A5,A5,$
FIELD=DENTIST_ZIP_LAST_FOUR,ALIAS=F0405_23,A4,A4,$
FIELD=DENTIST_LICENSE_NUMBER,ALIAS=F0405_24,A6,A6,$
FIELD=NUMBER_OF_PROCEDURES,ALIAS=F0405_25,I2,I2,$
FIELD=FIL 0001,ALIAS=F0405_26,A1,A1,$
FIELD=DENTIST_CHARGES,ALIAS=F0405_27,A256,A256,$
FIELD=AUTOPAD08001,ALIAS=,A256,A256,$
FIELD=AUTOPAD08002,ALIAS=,A256,A256,$
FIELD=AUTOPAD08003,ALIAS=,A32,A32,$
FIELD=S0405_08_KEY,ALIAS=DBKEY,I10,I4,$

```

```

SEGNAME=OCC08_01,SEGTYPE=S,PARENT=S0405_08,
    OCCURS=F0405_25,POSITION=F0405_27,$
FIELD=TOOTH_NUMBER,ALIAS=F0405_28,A2,A2,$
GROUP=SERVICE_DATE,ALIAS=F0405_29,A8,A8,$
FIELD=SERVICE_YEAR,ALIAS=F0405_30,A4,A4,$
FIELD=SERVICE_MONTH,ALIAS=F0405_31,A2,A2,$
FIELD=SERVICE_DAY,ALIAS=F0405_32,A2,A2,$
FIELD=PROCEDURE_CODE,ALIAS=F0405_33,A4,A4,$
FIELD=DESCRIPTION_OF_SERVICE,ALIAS=F0405_34,A60,A60,$
FIELD=FEE,ALIAS=F0405_35,P11.2,P5,$
FIELD=FIL 0001,ALIAS=F0405_36,A1,A1,$
FIELD=OCC08_01,ALIAS=ORDER,I9,I4,$

SEGNAME=S0430_09,SEGTYPE=S,PARENT=S0400_03,$
GROUP=CLAIM_DATE,ALIAS=F0430_1,A8,A8,$
FIELD=CLAIM_YEAR,ALIAS=F0430_2,A4,A4,$
FIELD=CLAIM_MONTH,ALIAS=F0430_3,A2,A2,$
FIELD=CLAIM_DAY,ALIAS=F0430_4,A2,A2,$
GROUP=PATIENT_NAME,ALIAS=F0430_5,A25,A25,$
FIELD=PATIENT_FIRST_NAME,ALIAS=F0430_6,A10,A10,$
FIELD=PATIENT_LAST_NAME,ALIAS=F0430_7,A15,A15,$
GROUP=PATIENT_BIRTH_DATE,ALIAS=F0430_8,A8,A8,$
FIELD=PATIENT_BIRTH_YEAR,ALIAS=F0430_9,A4,A4,$
FIELD=PATIENT_BIRTH_MONTH,ALIAS=F0430_10,A2,A2,$
FIELD=PATIENT_BIRTH_DAY,ALIAS=F0430_11,A2,A2,$
FIELD=PATIENT_SEX,ALIAS=F0430_12,A1,A1,$
FIELD=RELATION_TO_EMPLOYEE,ALIAS=F0430_13,A10,A10,$
FIELD=HOSPITAL_NAME,ALIAS=F0430_14,A25,A25,$
GROUP=HOSP_ADDRESS,ALIAS=F0430_15,A46,A46,$
FIELD=HOSP_STREET,ALIAS=F0430_16,A20,A20,$
FIELD=HOSP_CITY,ALIAS=F0430_17,A15,A15,$
FIELD=HOSP_STATE,ALIAS=F0430_18,A2,A2,$
GROUP=HOSP_ZIP,ALIAS=F0430_19,A9,A9,$
FIELD=HOSP_ZIP_FIRST_FIVE,ALIAS=F0430_20,A5,A5,$
FIELD=HOSP_ZIP_LAST_FOUR,ALIAS=F0430_21,A4,A4,$
GROUP=ADMIT_DATE,ALIAS=F0430_22,A8,A8,$
FIELD=ADMIT_YEAR,ALIAS=F0430_23,A4,A4,$
FIELD=ADMIT_MONTH,ALIAS=F0430_24,A2,A2,$
FIELD=ADMIT_DAY,ALIAS=F0430_25,A2,A2,$
GROUP=DISCHARGE_DATE,ALIAS=F0430_26,A8,A8,$
FIELD=DISCHARGE_YEAR,ALIAS=F0430_27,A4,A4,$
FIELD=DISCHARGE_MONTH,ALIAS=F0430_28,A2,A2,$
FIELD=DISCHARGE_DAY,ALIAS=F0430_29,A2,A2,$
FIELD=S09_OCC01,ALIAS=,A120,A120,$
GROUP=HOSPITAL_CHARGES,ALIAS=F0430_31,A72,A41,$
GROUP=ROOM_AND_BOARD,ALIAS=F0430_32,A48,A26,$
GROUP=WARD,ALIAS=F0430_33,A24,A13,$
FIELD=WARD_DAYS,ALIAS=F0430_34,P6,P3,$
FIELD=WARD_RATE,ALIAS=F0430_35,P11.2,P5,$
FIELD=WARD_TOTAL,ALIAS=F0430_36,P11.2,P5,$
GROUP=SEMI_PRIVATE,ALIAS=F0430_37,A24,A13,$
FIELD=SEMI_DAYS,ALIAS=F0430_38,P6,P3,$
FIELD=SEMI_RATE,ALIAS=F0430_39,P11.2,P5,$

```

```

FIELD=SEMI_TOTAL                ,ALIAS=F0430_40    ,P11.2 ,P5    ,,$
GROUP=OTHER_CHARGES             ,ALIAS=F0430_41    ,A24    ,A15    ,,$
FIELD=DELIVERY_COST             ,ALIAS=F0430_42    ,P11.2 ,P5    ,,$
FIELD=ANESTHESIA_COST          ,ALIAS=F0430_43    ,P11.2 ,P5    ,,$
FIELD=LAB_COST                  ,ALIAS=F0430_44    ,P11.2 ,P5    ,,$
FIELD=S0430_09_KEY              ,ALIAS=DBKEY       ,I10    ,I4     ,,$

SEGNAME=OCC09_01,SEGTYPE=S,PARENT=S0430_09,
OCCURS=0002,POSITION=S09_OCC01,$
FIELD=DIAGNOSIS                 ,ALIAS=F0430_30    ,A60    ,A60    ,,$
FIELD=OCC09_01                  ,ALIAS=ORDER       ,I9     ,I4     ,,$

SEGNAME=S0445_10,SEGTYPE=S,PARENT=S0400_03,$
GROUP=CLAIM_DATE                ,ALIAS=F0445_1     ,A8     ,A8     ,,$
FIELD=CLAIM_YEAR                ,ALIAS=F0445_2     ,A4     ,A4     ,,$
FIELD=CLAIM_MONTH               ,ALIAS=F0445_3     ,A2     ,A2     ,,$
FIELD=CLAIM_DAY                 ,ALIAS=F0445_4     ,A2     ,A2     ,,$
GROUP=PATIENT_NAME              ,ALIAS=F0445_5     ,A25    ,A25    ,,$
FIELD=PATIENT_FIRST_NAME        ,ALIAS=F0445_6     ,A10    ,A10    ,,$
FIELD=PATIENT_LAST_NAME         ,ALIAS=F0445_7     ,A15    ,A15    ,,$
GROUP=PATIENT_BIRTH_DATE        ,ALIAS=F0445_8     ,A8     ,A8     ,,$
FIELD=PATIENT_BIRTH_YEAR        ,ALIAS=F0445_9     ,A4     ,A4     ,,$
FIELD=PATIENT_BIRTH_MONTH       ,ALIAS=F0445_10    ,A2     ,A2     ,,$
FIELD=PATIENT_BIRTH_DAY         ,ALIAS=F0445_11    ,A2     ,A2     ,,$
FIELD=PATIENT_SEX               ,ALIAS=F0445_12    ,A1     ,A1     ,,$
FIELD=RELATION_TO_EMPLOYEE      ,ALIAS=F0445_13    ,A10    ,A10    ,,$
GROUP=PHYSICIAN_NAME            ,ALIAS=F0445_14    ,A25    ,A25    ,,$
FIELD=PHYSICIAN_FIRST_NAME      ,ALIAS=F0445_15    ,A10    ,A10    ,,$
FIELD=PHYSICIAN_LAST_NAME       ,ALIAS=F0445_16    ,A15    ,A15    ,,$
GROUP=PHYSICIAN_ADDRESS         ,ALIAS=F0445_17    ,A46    ,A46    ,,$
FIELD=PHYSICIAN_STREET          ,ALIAS=F0445_18    ,A20    ,A20    ,,$
FIELD=PHYSICIAN_CITY            ,ALIAS=F0445_19    ,A15    ,A15    ,,$
FIELD=PHYSICIAN_STATE           ,ALIAS=F0445_20    ,A2     ,A2     ,,$
GROUP=PHYSICIAN_ZIP             ,ALIAS=F0445_21    ,A9     ,A9     ,,$
FIELD=PHYSICIAN_ZIP_FIRST_FIVE  ,ALIAS=F0445_22    ,A5     ,A5     ,,$
FIELD=PHYSICIAN_ZIP_LAST_FOUR   ,ALIAS=F0445_23    ,A4     ,A4     ,,$
FIELD=PHYSICIAN_ID              ,ALIAS=F0445_24    ,A6     ,A6     ,,$
FIELD=S10_OCC01                 ,ALIAS=            ,A120   ,A120   ,,$
FIELD=NUMBER_OF_PROCEDURES      ,ALIAS=F0445_26    ,I2     ,I2     ,,$
FIELD=FIL 0001                  ,ALIAS=F0445_27    ,A1     ,A1     ,,$
FIELD=PHYSICIAN_CHARGES         ,ALIAS=F0445_28    ,A256   ,A256   ,,$
FIELD=AUTOPAD10001              ,ALIAS=            ,A256   ,A256   ,,$
FIELD=AUTOPAD10002              ,ALIAS=            ,A256   ,A256   ,,$
FIELD=AUTOPAD10003              ,ALIAS=            ,A32    ,A32    ,,$
FIELD=S0445_10_KEY              ,ALIAS=DBKEY       ,I10    ,I4     ,,$

SEGNAME=OCC10_01,SEGTYPE=S,PARENT=S0445_10,
OCCURS=0002,POSITION=S10_OCC01,$
FIELD=DIAGNOSIS                 ,ALIAS=F0445_25    ,A60    ,A60    ,,$
FIELD=OCC10_01                  ,ALIAS=ORDER       ,I9     ,I4     ,,$

```

```

SEGNAME=OCC10_02,SEGTYPE=S,PARENT=S0445_10,
    OCCURS=F0445_26,POSITION=F0445_28,$
    GROUP=SERVICE_DATE,ALIAS=F0445_29,A8,A8,$
    FIELD=SERVICE_YEAR,ALIAS=F0445_30,A4,A4,$
    FIELD=SERVICE_MONTH,ALIAS=F0445_31,A2,A2,$
    FIELD=SERVICE_DAY,ALIAS=F0445_32,A2,A2,$
    FIELD=PROCEDURE_CODE,ALIAS=F0445_33,A4,A4,$
    FIELD=DESCRIPTION_OF_SERVICE,ALIAS=F0445_34,A60,A60,$
    FIELD=FEE,ALIAS=F0445_35,P11.2,P5,$
    FIELD=FILE 0003,ALIAS=F0445_36,A3,A3,$
    FIELD=OCC10_02,ALIAS=ORDER,I9,I4,$

SEGNAME=S0440_11,SEGTYPE=U,PARENT=S0420_04,$
    FIELD=JOB_ID,ALIAS=F0440_1,A4,A4,$
    FIELD=TITLE,ALIAS=F0440_2,A20,A20,$
    GROUP=DESCRIPTION,ALIAS=F0440_3,A120,A120,$
    FIELD=S11_OCC01,ALIAS=,A120,A120,$
    GROUP=REQUIREMENTS,ALIAS=F0440_5,A120,A120,$
    FIELD=S11_OCC02,ALIAS=,A120,A120,$
    FIELD=MINIMUM_SALARY,ALIAS=F0440_7,P10.2,Z8,$
    FIELD=MAXIMUM_SALARY,ALIAS=F0440_8,P10.2,Z8,$
    FIELD=S11_OCC03,ALIAS=,A8,A8,$
    FIELD=NUMBER_OF_POSITIONS,ALIAS=F0440_10,A3,A3,$
    FIELD=NUMBER_OPEN,ALIAS=F0440_11,A3,A3,$
    FIELD=FILE 0002,ALIAS=F0440_12,A2,A2,$
    FIELD=S0440_11_KEY,ALIAS=DBKEY,I10,I4,$

SEGNAME=OCC11_01,SEGTYPE=S,PARENT=S0440_11,
    OCCURS=0002,POSITION=S11_OCC01,$
    FIELD=DESCRIPTION_LINE,ALIAS=F0440_4,A60,A60,$
    FIELD=OCC11_01,ALIAS=ORDER,I9,I4,$

SEGNAME=OCC11_02,SEGTYPE=S,PARENT=S0440_11,
    OCCURS=0002,POSITION=S11_OCC02,$
    FIELD=REQUIREMENT_LINE,ALIAS=F0440_6,A60,A60,$
    FIELD=OCC11_02,ALIAS=ORDER,I9,I4,$

SEGNAME=OCC11_03,SEGTYPE=S,PARENT=S0440_11,
    OCCURS=0004,POSITION=S11_OCC03,$
    FIELD=SALARY_GRADES,ALIAS=F0440_9,A2,A2,$
    FIELD=OCC11_03,ALIAS=ORDER,I9,I4,$

SEGNAME=S0455_12,SEGTYPE=U,PARENT=S0425_05,$
    FIELD=SKILL_ID,ALIAS=F0455_1,A4,A4,$
    FIELD=SKILL_NAME,ALIAS=F0455_2,A12,A12,$
    FIELD=SKILL_DESCRIPTION,ALIAS=F0455_3,A60,A60,$
    FIELD=S0455_12_KEY,ALIAS=DBKEY,I10,I4,$

```

```

SEGNAME=S0415_13,SEGTYPE=U,PARENT=S0460_06,$
FIELD=EMP_ID,ALIAS=F0415_1A,A4,A4,$
GROUP=EMP_NAME,ALIAS=F0415_2A,A25,A25,$
FIELD=EMP_FIRST_NAME,ALIAS=F0415_3A,A10,A10,$
FIELD=EMP_LAST_NAME,ALIAS=F0415_4A,A15,A15,$
GROUP=EMP_ADDRESS,ALIAS=F0415_5A,A46,A46,$
FIELD=EMP_STREET,ALIAS=F0415_6A,A20,A20,$
FIELD=EMP_CITY,ALIAS=F0415_7A,A15,A15,$
FIELD=EMP_STATE,ALIAS=F0415_8A,A2,A2,$
GROUP=EMP_ZIP,ALIAS=F0415_9A,A9,A9,$
FIELD=EMP_ZIP_FIRST_FIVE,ALIAS=F0415_10A,A5,A5,$
FIELD=EMP_ZIP_LAST_FOUR,ALIAS=F0415_11A,A4,A4,$
FIELD=EMP_PHONE,ALIAS=F0415_12A,A10,A10,$
FIELD=STATUS,ALIAS=F0415_13A,A2,A2,$
FIELD=SS_NUMBER,ALIAS=F0415_19A,A9,A9,$
GROUP=START_DATE,ALIAS=F0415_20A,A8,A8,$
FIELD=START_YEAR,ALIAS=F0415_21A,A4,A4,$
FIELD=START_MONTH,ALIAS=F0415_22A,A2,A2,$
FIELD=START_DAY,ALIAS=F0415_23A,A2,A2,$
GROUP=TERMINATION_DATE,ALIAS=F0415_24A,A8,A8,$
FIELD=TERMINATION_YEAR,ALIAS=F0415_25A,A4,A4,$
FIELD=TERMINATION_MONTH,ALIAS=F0415_26A,A2,A2,$
FIELD=TERMINATION_DAY,ALIAS=F0415_27A,A2,A2,$
GROUP=BIRTH_DATE,ALIAS=F0415_28A,A8,A8,$
FIELD=BIRTH_YEAR,ALIAS=F0415_29A,A4,A4,$
FIELD=BIRTH_MONTH,ALIAS=F0415_30A,A2,A2,$
FIELD=BIRTH_DAY,ALIAS=F0415_31A,A2,A2,$
FIELD=S0415_13_KEY,ALIAS=DBKEY,I10,I4,$
$DUPLICATE=BIRTH_DATE,COUNT= 2,SEGNAME=S0415_02
$DUPLICATE=BIRTH_DATE,COUNT= 2,SEGNAME=S0415_13
$DUPLICATE=BIRTH_DAY,COUNT= 2,SEGNAME=S0415_02
$DUPLICATE=BIRTH_DAY,COUNT= 2,SEGNAME=S0415_13
$DUPLICATE=BIRTH_MONTH,COUNT= 2,SEGNAME=S0415_02
$DUPLICATE=BIRTH_MONTH,COUNT= 2,SEGNAME=S0415_13
$DUPLICATE=BIRTH_YEAR,COUNT= 2,SEGNAME=S0415_02
$DUPLICATE=BIRTH_YEAR,COUNT= 2,SEGNAME=S0415_13
$DUPLICATE=CLAIM_DATE,COUNT= 3,SEGNAME=S0405_08
$DUPLICATE=CLAIM_DATE,COUNT= 3,SEGNAME=S0430_09
$DUPLICATE=CLAIM_DATE,COUNT= 3,SEGNAME=S0445_10
$DUPLICATE=CLAIM_DAY,COUNT= 3,SEGNAME=S0405_08
$DUPLICATE=CLAIM_DAY,COUNT= 3,SEGNAME=S0430_09
$DUPLICATE=CLAIM_DAY,COUNT= 3,SEGNAME=S0445_10
$DUPLICATE=CLAIM_MONTH,COUNT= 3,SEGNAME=S0405_08
$DUPLICATE=CLAIM_MONTH,COUNT= 3,SEGNAME=S0430_09
$DUPLICATE=CLAIM_MONTH,COUNT= 3,SEGNAME=S0445_10
$DUPLICATE=CLAIM_YEAR,COUNT= 3,SEGNAME=S0405_08
$DUPLICATE=CLAIM_YEAR,COUNT= 3,SEGNAME=S0430_09
$DUPLICATE=CLAIM_YEAR,COUNT= 3,SEGNAME=S0445_10
$DUPLICATE=DESCRIPTION_OF_SERVICE,COUNT= 2,SEGNAME=S0405_08
$DUPLICATE=DESCRIPTION_OF_SERVICE,COUNT= 2,SEGNAME=S0445_10
$DUPLICATE=DIAGNOSIS,COUNT= 2,SEGNAME=S0430_09
$DUPLICATE=DIAGNOSIS,COUNT= 2,SEGNAME=S0445_10
$DUPLICATE=EMP_ADDRESS,COUNT= 2,SEGNAME=S0415_02

```

\$DUPLICATE=EMP_ADDRESS	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=EMP_CITY	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=EMP_CITY	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=EMP_FIRST_NAME	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=EMP_FIRST_NAME	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=EMP_ID	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=EMP_ID	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=EMP_LAST_NAME	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=EMP_LAST_NAME	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=EMP_NAME	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=EMP_NAME	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=EMP_PHONE	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=EMP_PHONE	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=EMP_STATE	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=EMP_STATE	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=EMP_STREET	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=EMP_STREET	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=EMP_ZIP	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=EMP_ZIP	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=EMP_ZIP_FIRST_FIVE	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=EMP_ZIP_FIRST_FIVE	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=EMP_ZIP_LAST_FOUR	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=EMP_ZIP_LAST_FOUR	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=FEE	,COUNT= 2,SEGNAME=S0405_08
\$DUPLICATE=FEE	,COUNT= 2,SEGNAME=S0445_10
\$DUPLICATE=FIL 0001	,COUNT= 3,SEGNAME=S0405_08
\$DUPLICATE=FIL 0001	,COUNT= 3,SEGNAME=S0405_08
\$DUPLICATE=FIL 0001	,COUNT= 3,SEGNAME=S0445_10
\$DUPLICATE=FIL 0002	,COUNT= 3,SEGNAME=S0425_05
\$DUPLICATE=FIL 0002	,COUNT= 3,SEGNAME=S0460_06
\$DUPLICATE=FIL 0002	,COUNT= 3,SEGNAME=S0440_11
\$DUPLICATE=FIL 0003	,COUNT= 3,SEGNAME=S0410_01
\$DUPLICATE=FIL 0003	,COUNT= 3,SEGNAME=S0420_04
\$DUPLICATE=FIL 0003	,COUNT= 3,SEGNAME=S0445_10
\$DUPLICATE=NUMBER_OF_PROCEDURES	,COUNT= 2,SEGNAME=S0405_08
\$DUPLICATE=NUMBER_OF_PROCEDURES	,COUNT= 2,SEGNAME=S0445_10
\$DUPLICATE=PATIENT_BIRTH_DATE	,COUNT= 3,SEGNAME=S0405_08
\$DUPLICATE=PATIENT_BIRTH_DATE	,COUNT= 3,SEGNAME=S0430_09
\$DUPLICATE=PATIENT_BIRTH_DATE	,COUNT= 3,SEGNAME=S0445_10
\$DUPLICATE=PATIENT_BIRTH_DAY	,COUNT= 3,SEGNAME=S0405_08
\$DUPLICATE=PATIENT_BIRTH_DAY	,COUNT= 3,SEGNAME=S0430_09
\$DUPLICATE=PATIENT_BIRTH_DAY	,COUNT= 3,SEGNAME=S0445_10
\$DUPLICATE=PATIENT_BIRTH_MONTH	,COUNT= 3,SEGNAME=S0405_08
\$DUPLICATE=PATIENT_BIRTH_MONTH	,COUNT= 3,SEGNAME=S0430_09
\$DUPLICATE=PATIENT_BIRTH_MONTH	,COUNT= 3,SEGNAME=S0445_10
\$DUPLICATE=PATIENT_BIRTH_YEAR	,COUNT= 3,SEGNAME=S0405_08
\$DUPLICATE=PATIENT_BIRTH_YEAR	,COUNT= 3,SEGNAME=S0430_09
\$DUPLICATE=PATIENT_BIRTH_YEAR	,COUNT= 3,SEGNAME=S0445_10
\$DUPLICATE=PATIENT_FIRST_NAME	,COUNT= 3,SEGNAME=S0405_08
\$DUPLICATE=PATIENT_FIRST_NAME	,COUNT= 3,SEGNAME=S0430_09
\$DUPLICATE=PATIENT_FIRST_NAME	,COUNT= 3,SEGNAME=S0445_10
\$DUPLICATE=PATIENT_LAST_NAME	,COUNT= 3,SEGNAME=S0405_08
\$DUPLICATE=PATIENT_LAST_NAME	,COUNT= 3,SEGNAME=S0430_09

\$DUPLICATE=PATIENT_LAST_NAME	,COUNT= 3,SEGNAME=S0445_10
\$DUPLICATE=PATIENT_NAME	,COUNT= 3,SEGNAME=S0405_08
\$DUPLICATE=PATIENT_NAME	,COUNT= 3,SEGNAME=S0430_09
\$DUPLICATE=PATIENT_NAME	,COUNT= 3,SEGNAME=S0445_10
\$DUPLICATE=PATIENT_SEX	,COUNT= 3,SEGNAME=S0405_08
\$DUPLICATE=PATIENT_SEX	,COUNT= 3,SEGNAME=S0430_09
\$DUPLICATE=PATIENT_SEX	,COUNT= 3,SEGNAME=S0445_10
\$DUPLICATE=PROCEDURE_CODE	,COUNT= 2,SEGNAME=S0405_08
\$DUPLICATE=PROCEDURE_CODE	,COUNT= 2,SEGNAME=S0445_10
\$DUPLICATE=RELATION_TO_EMPLOYEE	,COUNT= 3,SEGNAME=S0405_08
\$DUPLICATE=RELATION_TO_EMPLOYEE	,COUNT= 3,SEGNAME=S0430_09
\$DUPLICATE=RELATION_TO_EMPLOYEE	,COUNT= 3,SEGNAME=S0445_10
\$DUPLICATE=SERVICE_DATE	,COUNT= 2,SEGNAME=S0405_08
\$DUPLICATE=SERVICE_DATE	,COUNT= 2,SEGNAME=S0445_10
\$DUPLICATE=SERVICE_DAY	,COUNT= 2,SEGNAME=S0405_08
\$DUPLICATE=SERVICE_DAY	,COUNT= 2,SEGNAME=S0445_10
\$DUPLICATE=SERVICE_MONTH	,COUNT= 2,SEGNAME=S0405_08
\$DUPLICATE=SERVICE_MONTH	,COUNT= 2,SEGNAME=S0445_10
\$DUPLICATE=SERVICE_YEAR	,COUNT= 2,SEGNAME=S0405_08
\$DUPLICATE=SERVICE_YEAR	,COUNT= 2,SEGNAME=S0445_10
\$DUPLICATE=SS_NUMBER	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=SS_NUMBER	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=START_DATE	,COUNT= 3,SEGNAME=S0415_02
\$DUPLICATE=START_DATE	,COUNT= 3,SEGNAME=S0420_04
\$DUPLICATE=START_DATE	,COUNT= 3,SEGNAME=S0415_13
\$DUPLICATE=START_DAY	,COUNT= 3,SEGNAME=S0415_02
\$DUPLICATE=START_DAY	,COUNT= 3,SEGNAME=S0420_04
\$DUPLICATE=START_DAY	,COUNT= 3,SEGNAME=S0415_13
\$DUPLICATE=START_MONTH	,COUNT= 3,SEGNAME=S0415_02
\$DUPLICATE=START_MONTH	,COUNT= 3,SEGNAME=S0420_04
\$DUPLICATE=START_MONTH	,COUNT= 3,SEGNAME=S0415_13
\$DUPLICATE=START_YEAR	,COUNT= 3,SEGNAME=S0415_02
\$DUPLICATE=START_YEAR	,COUNT= 3,SEGNAME=S0420_04
\$DUPLICATE=START_YEAR	,COUNT= 3,SEGNAME=S0415_13
\$DUPLICATE=STATUS	,COUNT= 2,SEGNAME=S0415_02
\$DUPLICATE=STATUS	,COUNT= 2,SEGNAME=S0415_13
\$DUPLICATE=TERMINATION_DATE	,COUNT= 3,SEGNAME=S0415_02
\$DUPLICATE=TERMINATION_DATE	,COUNT= 3,SEGNAME=S0400_03
\$DUPLICATE=TERMINATION_DATE	,COUNT= 3,SEGNAME=S0415_13
\$DUPLICATE=TERMINATION_DAY	,COUNT= 3,SEGNAME=S0415_02
\$DUPLICATE=TERMINATION_DAY	,COUNT= 3,SEGNAME=S0400_03
\$DUPLICATE=TERMINATION_DAY	,COUNT= 3,SEGNAME=S0415_13
\$DUPLICATE=TERMINATION_MONTH	,COUNT= 3,SEGNAME=S0415_02
\$DUPLICATE=TERMINATION_MONTH	,COUNT= 3,SEGNAME=S0400_03
\$DUPLICATE=TERMINATION_MONTH	,COUNT= 3,SEGNAME=S0415_13
\$DUPLICATE=TERMINATION_YEAR	,COUNT= 3,SEGNAME=S0415_02
\$DUPLICATE=TERMINATION_YEAR	,COUNT= 3,SEGNAME=S0400_03
\$DUPLICATE=TERMINATION_YEAR	,COUNT= 3,SEGNAME=S0415_13

EMPFULL Access File

This Access File is used in the examples throughout this manual. For instructions about creating this Access File, see the sample session in Chapter 6, *Creating File Descriptions With AUTOIDMS*:

```

$$$ CREATED BY AUTOIDMS ON 02/13/01 AT 14.32.16 BY PMSSAE
$$$ FILE=EMPFULL,SUFFIX=IDMSR,$
SSHEMA=EMPSS01,RELEASE=12.0,MODE=DML,
DBNAME=EMPDEMO ,DICTNAME=APPLDICT , $

SEGNAM=S0410_01,RECORD=DEPARTMENT,
      AREA=ORG-DEMO-REGION,
CLCFLD=F0410_1,
      CLCDUP=N,$

SEGNAM=S0415_02,RECORD=EMPLOYEE,
      AREA=EMP-DEMO-REGION,
      ACCESS=SET,SETNAME=DEPT-EMPLOYEE,
      SETMBR=OA,GETOWN=Y,MULTMBR=N,
CLCFLD=F0415_1,
      CLCDUP=N,
KEYFLD=F0415_4/F0415_3,
      SETORD=A,SETDUP=Y,$
IXSET=EMP-NAME-NDX,IXDUP=Y,IXORD=A,
IXFLD=F0415_4/F0415_3,$

SEGNAM=S0400_03,RECORD=COVERAGE,
      AREA=INS-DEMO-REGION,
      ACCESS=SET,SETNAME=EMP-COVERAGE,
      SETMBR=MA,GETOWN=Y,MULTMBR=N,$

SEGNAM=S0420_04,RECORD=EMPOSITION,
      AREA=EMP-DEMO-REGION,
      ACCESS=SET,SETNAME=EMP-EMPOSITION,
      SETMBR=MA,GETOWN=Y,MULTMBR=N,$

SEGNAM=S0425_05,RECORD=EXPERTISE,
      AREA=EMP-DEMO-REGION,
      ACCESS=SET,SETNAME=EMP-EXPERTISE,
      SETMBR=MA,GETOWN=Y,MULTMBR=N,
KEYFLD=F0425_1,
      SETORD=D,SETDUP=Y,$

SEGNAM=S0460_06,RECORD=STRUCTURE,
      AREA=EMP-DEMO-REGION,
      ACCESS=SET,SETNAME=MANAGES,
      SETMBR=MA,GETOWN=Y,MULTMBR=N,$

```



```
SEGNAM=S0450_07,RECORD=OFFICE,
      AREA=ORG-DEMO-REGION,
      ACCESS=SET,SETNAME=OFFICE-EMPLOYEE,
      SETMBR=OA,GETOWN=Y,MULTMBR=N,
CLCFLD=F0450_1,
      CLCDUP=N,
KEYFLD=F0415_4/F0415_3,
      SETORD=A,SETDUP=Y,$

SEGNAM=S0405_08,RECORD=DENTAL-CLAIM,
      AREA=INS-DEMO-REGION,
      ACCESS=SET,SETNAME=COVERAGE-CLAIMS,
      SETMBR=MA,GETOWN=N,MULTMBR=Y,$

SEGNAM=S0430_09,RECORD=HOSPITAL-CLAIM,
      AREA=INS-DEMO-REGION,
      ACCESS=SET,SETNAME=COVERAGE-CLAIMS,
      SETMBR=MA,GETOWN=N,MULTMBR=Y,$

SEGNAM=S0445_10,RECORD=NON-HOSP-CLAIM,
      AREA=INS-DEMO-REGION,
      ACCESS=SET,SETNAME=COVERAGE-CLAIMS,
      SETMBR=MA,GETOWN=N,MULTMBR=Y,$

SEGNAM=S0440_11,RECORD=JOB,
      AREA=ORG-DEMO-REGION,
      ACCESS=SET,SETNAME=JOB-EMPOSITION,
      SETMBR=OM,GETOWN=Y,MULTMBR=N,
CLCFLD=F0440_1,
      CLCDUP=N,$
IXSET=JOB-TITLE-NDX,IXDUP=N,IXORD=A,
IXFLD=F0440_2,$

SEGNAM=S0455_12,RECORD=SKILL,
      AREA=ORG-DEMO-REGION,
      ACCESS=SET,SETNAME=SKILL-EXPERTISE,
      SETMBR=MA,GETOWN=Y,MULTMBR=N,
CLCFLD=F0455_1,
      CLCDUP=N,
KEYFLD=F0425_1,
      SETORD=D,SETDUP=Y,$
IXSET=SKILL-NAME-NDX,IXDUP=N,IXORD=A,
IXFLD=F0455_2,$

SEGNAM=S0415_13,RECORD=EMPLOYEE,
      AREA=EMP-DEMO-REGION,
      ACCESS=SET,SETNAME=REPORTS-TO,
      SETMBR=OM,GETOWN=Y,MULTMBR=N,
CLCFLD=F0415_1A,
      CLCDUP=N,$
IXSET=EMP-NAME-NDX,IXDUP=Y,IXORD=A,
IXFLD=F0415_4A/F0415_3A,$
```

EMPFILE Master File

This Master File is used in the trace examples shown in Appendix C, *Tracing CA-IDMS Data Adapter Processing*. For instructions about creating this Master File, see the sample session in Chapter 6, *Creating File Descriptions With AUTOIDMS*:

```
$$$ CREATED BY AUTOIDMS ON 02/20/01 AT 11.52.57 BY PMSSAE
$$$ SSHEMA=EMPSS01
FILE=EMPFILE,SUFFIX=IDMSR,$
```

```
SEGNAME=S0415_01,SEGTYPE=S,$
```

FIELD=EMP_ID	,ALIAS=F0415_1	,A4	,A4	,\$
GROUP=EMP_NAME	,ALIAS=F0415_2	,A25	,A25	,\$
FIELD=EMP_FIRST_NAME	,ALIAS=F0415_3	,A10	,A10	,\$
FIELD=EMP_LAST_NAME	,ALIAS=F0415_4	,A15	,A15	,\$
GROUP=EMP_ADDRESS	,ALIAS=F0415_5	,A46	,A46	,\$
FIELD=EMP_STREET	,ALIAS=F0415_6	,A20	,A20	,\$
FIELD=EMP_CITY	,ALIAS=F0415_7	,A15	,A15	,\$
FIELD=EMP_STATE	,ALIAS=F0415_8	,A2	,A2	,\$
GROUP=EMP_ZIP	,ALIAS=F0415_9	,A9	,A9	,\$
FIELD=EMP_ZIP_FIRST_FIVE	,ALIAS=F0415_10	,A5	,A5	,\$
FIELD=EMP_ZIP_LAST_FOUR	,ALIAS=F0415_11	,A4	,A4	,\$
FIELD=EMP_PHONE	,ALIAS=F0415_12	,A10	,A10	,\$
FIELD=STATUS	,ALIAS=F0415_13	,A2	,A2	,\$
FIELD=SS_NUMBER	,ALIAS=F0415_19	,A9	,A9	,\$
GROUP=START_DATE	,ALIAS=F0415_20	,A8	,A8	,\$
FIELD=START_YEAR	,ALIAS=F0415_21	,A4	,A4	,\$
FIELD=START_MONTH	,ALIAS=F0415_22	,A2	,A2	,\$
FIELD=START_DAY	,ALIAS=F0415_23	,A2	,A2	,\$
GROUP=TERMINATION_DATE	,ALIAS=F0415_24	,A8	,A8	,\$
FIELD=TERMINATION_YEAR	,ALIAS=F0415_25	,A4	,A4	,\$
FIELD=TERMINATION_MONTH	,ALIAS=F0415_26	,A2	,A2	,\$
FIELD=TERMINATION_DAY	,ALIAS=F0415_27	,A2	,A2	,\$
GROUP=BIRTH_DATE	,ALIAS=F0415_28	,A8	,A8	,\$
FIELD=BIRTH_YEAR	,ALIAS=F0415_29	,A4	,A4	,\$
FIELD=BIRTH_MONTH	,ALIAS=F0415_30	,A2	,A2	,\$
FIELD=BIRTH_DAY	,ALIAS=F0415_31	,A2	,A2	,\$
FIELD=S0415_01_KEY	,ALIAS=DBKEY	,I10	,I4	,\$

```
SEGNAME=S0410_02,SEGTYPE=U,PARENT=S0415_01,$
```

FIELD=DEPT_ID	,ALIAS=F0410_1	,A4	,A4	,\$
FIELD=DEPT_NAME	,ALIAS=F0410_2	,A45	,A45	,\$
FIELD=DEPT_HEAD_ID	,ALIAS=F0410_3	,A4	,A4	,\$
FIELD=FIL 0003	,ALIAS=F0410_4	,A3	,A3	,\$
FIELD=S0410_02_KEY	,ALIAS=DBKEY	,I10	,I4	,\$

EMPFIL Access File

This Access File is used in the trace examples shown in Appendix C, *Tracing CA-IDMS Data Adapter Processing*. For instructions about creating this Access File, see the sample session in Chapter 6, *Creating File Descriptions With AUTOIDMS*:

```
$$$ CREATED BY AUTOIDMS ON 02/20/01 AT 11.52.57 BY PMSSAE
$$$ FILE=EMPFIL,SUFFIX=IDMSR,$
SSHEMA=EMPSS01,RELEASE=12.0,MODE=DML,
DBNAME=EMPDEMO ,DICTNAME=APPLDICT , $

SEGNAM=S0415_01,RECORD=EMPLOYEE,
      AREA=EMP-DEMO-REGION,
CLCFLD=F0415_1,
      CLCDUP=N,$
IXSET=EMP-NAME-NDX,IXDUP=Y,IXORD=A,
IXFLD=F0415_4/F0415_3,$

SEGNAM=S0410_02,RECORD=DEPARTMENT,
      AREA=ORG-DEMO-REGION,
      ACCESS=SET,SETNAME=DEPT-EMPLOYEE,
      SETMBR=OA,GETOWN=Y,MULTMBR=N,
CLCFLD=F0410_1,
      CLCDUP=N,
KEYFLD=F0415_4/F0415_3,
      SETORD=A,SETDUP=Y,$
```

Sample CA-IDMS Report

IDMSRPTS 14.1	----- INPUT PARAMETER LISTING -----				DATE	TIME	PAGE					
	DICTIONARY	DEFAULT	OF	NODE	DEFAULT	02/13/01	142430 1					
	SCHEMA=EMPSCHM		VERSION=HIGH									
	RECDES											

	SCHEMA	VERSION	SUBSCHEMA	REPORTS								
	EMPSCHM	HIGHEST		RECDES								
IDMSRPTS 14.1	-- SCHEMA RECORD DESCRIPTION LISTING ---				DATE	TIME	PAGE					
RECDES	DICTIONARY	DEFAULT	OF	NODE	DEFAULT	02/13/01	142430 1					
			SCHEMA	EMPSCHM	VERSION	100						
RECORD NAME.....	COVERAGE					RLGTH=	40					
RECORD VERSION.....	0100					DLGTH=	20					
RECORD ID.....	0400					KLGTH=	20					
RECORD LENGTH.....	FIXED					DSTRT=	20					
LOCATION MODE.....	VIA SET	EMP-COVERAGE	DISPLACEMENT 0000		PAGES							
WITHIN.....	INS-DEMO-REGION	OFFSET	5 PGS FOR		45 PGS							
DBKEY POSITIONS....	SET.....	TYPE.....	NEXT	PRIOR	OWNER							
	EMP-COVERAGE	MEMBER	1	2	3							
	COVERAGE-CLAIMS	OWNER	4	5								
DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....	STRT	LGTH						
02 SELECTION-DATE-0400		DISPLAY			1	8						
03 SELECTION-YEAR-0400		DISPLAY		9(4)	1	4						
03 SELECTION-MONTH-0400		DISPLAY		9(2)	5	2						
03 SELECTION-DAY-0400		DISPLAY		9(2)	7	2						
02 TERMINATION-DATE-0400		DISPLAY			9	8						
03 TERMINATION-YEAR-0400		DISPLAY		9(4)	9	4						
03 TERMINATION-MONTH-0400		DISPLAY		9(2)	13	2						
03 TERMINATION-DAY-0400		DISPLAY		9(2)	15	2						
02 TYPE-0400		DISPLAY		X	17	1						
88 MASTER-0400		COND	'M'		17							
88 FAMILY-0400		COND	'F'		17							
88 DEPENDENT-0400		COND	'D'		17							
02 INS-PLAN-CODE-0400		DISPLAY		X(3)	18	3						
88 GROUP-LIFE-0400		COND	'001'		18							
88 HMO-0400		COND	'002'		18							
88 GROUP-HEALTH-0400		COND	'003'		18							
88 GROUP-DENTAL-0400		COND	'004'		18							
*****				*****								

```
REC SYNONYM NAME... COVERGE                                COVERAGE                                RLGTH=   40
REC SYNONYM VER.... 0100                                DLGTH=   20
LANGUAGE(S)..... ASSEMBLER
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....  STRT  LGTH

02 COVSELDT                                DISPLAY                                1      8
03 COVSELYR                                DISPLAY                                9(4)    1      4
03 COVSELMO                                DISPLAY                                9(2)    5      2
03 COVSELDA                                DISPLAY                                9(2)    7      2
02 COVTRMDT                                DISPLAY                                9      8
03 COVTRMYR                                DISPLAY                                9(4)    9      4
03 COVTRMMO                                DISPLAY                                9(2)   13      2
03 COVTRMDA                                DISPLAY                                9(2)   15      2
02 COVTYPE                                DISPLAY                                X      17      1
88 COVMASTR                                COND                                'M'    17
88 COVFAMLY                                COND                                'F'    17
88 COVDPNDT                                COND                                'D'    17
IDMSRPTS 14.1                                -- SCHEMA RECORD DESCRIPTION LISTING ---          DATE    TIME    PAGE
RECDES                                DICTIONARY DEFAULT OF NODE DEFAULT          02/13/01 142430    2
                                SCHEMA EMPSCHM VERSION 100

DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....  STRT  LGTH

02 COVPLNCD                                DISPLAY                                X(3)   18      3
88 GROUP~LIFE                                COND                                '001'  18
88 HMO                                COND                                '002'  18
88 GROUP~HEALTH                                COND                                '003'  18
88 GROUP~DENTAL                                COND                                '004'  18

*****                                *****

REC SYNONYM NAME... COVERGE                                COVERAGE                                RLGTH=   40
REC SYNONYM VER.... 0100                                DLGTH=   20
LANGUAGE(S)..... FORTRAN
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....  STRT  LGTH

02 CVSLDT                                DISPLAY                                1      8
03 CVSLYR                                DISPLAY                                9(4)    1      4
03 CVSLMO                                DISPLAY                                9(2)    5      2
03 CVSLDA                                DISPLAY                                9(2)    7      2
02 CVTMDT                                DISPLAY                                9      8
03 CVTMYR                                DISPLAY                                9(4)    9      4
03 CVTMMO                                DISPLAY                                9(2)   13      2
03 CVTMDA                                DISPLAY                                9(2)   15      2
02 CVTYPE                                DISPLAY                                X      17      1
88 CVMSTR                                COND                                'M'    17
88 CVFMLY                                COND                                'F'    17
88 CVDPND                                COND                                'D'    17
02 CVPLCD                                DISPLAY                                X(3)   18      3
88 GROUP~LIFE                                COND                                '001'  18
88 HMO                                COND                                '002'  18
88 GROUP~HEALTH                                COND                                '003'  18
88 GROUP~DENTAL                                COND                                '004'  18
```

```

IDMSRPTS 14.1          -- SCHEMA RECORD DESCRIPTION LISTING ---          DATE      TIME      PAGE
RECDES                DICTIONARY DEFAULT  OF NODE DEFAULT          02/13/01  142430      3
                        SCHEMA EMPSCHM  VERSION 100

RECORD NAME..... DENTAL-CLAIM                                RLGTH=  948
RECORD VERSION..... 0100                                DLGTH=  936
RECORD ID..... 0405                                KLGTH=   12
RECORD LENGTH..... VARIABLE                                DSTRT=   16
MINIMUM ROOT..... 132  CHARACTERS
MINIMUM FRAGMENT... 932  CHARACTERS
LOCATION MODE..... VIA SET      COVERAGE-CLAIMS      DISPLACEMENT 0000  PAGES
WITHIN..... INS-DEMO-REGION      OFFSET            5 PGS FOR      45 PGS
DBKEY POSITIONS.... SET..... TYPE..... NEXT  PRIOR OWNER
                        COVERAGE-CLAIMS  MEMBER            1      2
                        (FRAGMENT CHAIN) INTRNL            3
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....  STRT  LGTH

02 CLAIM-DATE-0405          DISPLAY                                1      8
03 CLAIM-YEAR-0405          DISPLAY                                9(4)    1      4
03 CLAIM-MONTH-0405         DISPLAY                                9(2)    5      2
03 CLAIM-DAY-0405           DISPLAY                                9(2)    7      2
02 PATIENT-NAME-0405        DISPLAY                                9      25
03 PATIENT-FIRST-NAME-0405   DISPLAY                                X(10)   9     10
03 PATIENT-LAST-NAME-0405    DISPLAY                                X(15)  19     15
02 PATIENT-BIRTH-DATE-0405   DISPLAY                                34      8
03 PATIENT-BIRTH-YEAR-0405   DISPLAY                                9(4)   34      4
03 PATIENT-BIRTH-MONTH-0405  DISPLAY                                9(2)   38      2
03 PATIENT-BIRTH-DAY-0405    DISPLAY                                9(2)   40      2
02 PATIENT-SEX-0405          DISPLAY                                X      42      1
02 RELATION-TO-EMPLOYEE-0405 DISPLAY                                X(10)  43     10
02 DENTIST-NAME-0405         DISPLAY                                53     25
03 DENTIST-FIRST-NAME-0405    DISPLAY                                X(10)  53     10
03 DENTIST-LAST-NAME-0405     DISPLAY                                X(15)  63     15
02 DENTIST-ADDRESS-0405      DISPLAY                                78     46
03 DENTIST-STREET-0405       DISPLAY                                X(20)  78     20
03 DENTIST-CITY-0405         DISPLAY                                X(15)  98     15
03 DENTIST-STATE-0405        DISPLAY                                X(2)   113     2
03 DENTIST-ZIP-0405          DISPLAY                                115     9
04 DENTIST-ZIP-FIRST-FIVE-0405 DISPLAY                                X(5)   115     5
04 DENTIST-ZIP-LAST-FOUR-0405 DISPLAY                                X(4)   120     4
02 DENTIST-LICENSE-NUMBER-0405 DISPLAY                                9(6)   124     6
02 NUMBER-OF-PROCEDURES-0405 COMP                                9(2)   130     2
02 FILLER                   DISPLAY                                X      132     1
02 DENTIST-CHARGES-0405      DISPLAY      OCCURS 0 TO    10                                133    800
                        DEPENDING ON -----      NUMBER-OF-PROCEDURES-0405
03 TOOTH-NUMBER-0405         DISPLAY                                9(2)    1      2
03 SERVICE-DATE-0405         DISPLAY                                3      3      8
04 SERVICE-YEAR-0405         DISPLAY                                9(4)    3      4
04 SERVICE-MONTH-0405        DISPLAY                                9(2)    7      2
04 SERVICE-DAY-0405          DISPLAY                                9(2)    9      2
03 PROCEDURE-CODE-0405       DISPLAY                                9(4)   11      4
03 DESCRIPTION-OF-SERVICE-0405 DISPLAY                                X(60)   15     60
03 FEE-0405                  COMP-3                                S9(7)V99 75      5

```

```
03 FILLER                                DISPLAY                                X                                80      1
IDMSRPTS 14.1                          -- SCHEMA RECORD DESCRIPTION LISTING ---          DATE      TIME      PAGE
RECDES                                DICTIONARY DEFAULT  OF NODE DEFAULT          02/13/01  142430      4
                                SCHEMA EMPSCHM  VERSION  100

REC SYNONYM NAME... DENTCLM                                DENTAL-CLAIM                                RLGTH=  948
REC SYNONYM VER.... 0100                                DLGTH=  936
LANGUAGE(S)..... ASSEMBLER
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....      STRT  LGTH

02 DCCLMDT                                DISPLAY                                1        8
03 DCCLMYR                                DISPLAY                                9(4)      1        4
03 DCCLMMO                                DISPLAY                                9(2)      5        2
03 DCCLMDA                                DISPLAY                                9(2)      7        2
02 DCPNAME                                DISPLAY                                9        25
03 DCPFNAME                                DISPLAY                                X(10)     9        10
03 DCPINAME                                DISPLAY                                X(15)    19        15
02 DCPBIRDT                                DISPLAY                                34        8
03 DCPBIRYR                                DISPLAY                                9(4)     34        4
03 DCPBIRMO                                DISPLAY                                9(2)     38        2
03 DCPBIRDA                                DISPLAY                                9(2)     40        2
02 DCPSEX                                DISPLAY                                X        42        1
02 DCRELEMP                                DISPLAY                                X(10)    43        10
02 DCDNNAME                                DISPLAY                                53        25
03 DCDNFNAM                                DISPLAY                                X(10)    53        10
03 DCDNLNAM                                DISPLAY                                X(15)    63        15
02 DCDNADDR                                DISPLAY                                78        46
03 DCDNSTR                                DISPLAY                                X(20)    78        20
03 DCDNCITY                                DISPLAY                                X(15)    98        15
03 DCDNSTAT                                DISPLAY                                X(2)    113        2
03 DCDNZIP                                DISPLAY                                115        9
04 DCDNZPF5                                DISPLAY                                X(5)    115        5
04 DCDNZPL4                                DISPLAY                                X(4)    120        4
02 DCDNLICN                                DISPLAY                                9(6)    124        6
02 DCNOPROC                                COMP                                9(2)    130        2
02 FILLER                                DISPLAY                                X        132        1
02 DCDNCHGS                                DISPLAY      OCCURS 0 TO    10          133    800
                                DEPENDING ON -----          DCNOPROC
03 DCTOTHNO                                DISPLAY                                9(2)      1        2
03 DCSERVDT                                DISPLAY                                3        8
04 DCSERVYR                                DISPLAY                                9(4)      3        4
04 DCSERVMO                                DISPLAY                                9(2)      7        2
04 DCSERVDA                                DISPLAY                                9(2)      9        2
03 DCPROCCD                                DISPLAY                                9(4)     11        4
03 DCDDESCSV                                DISPLAY                                X(60)    15        60
03 DCFEE                                COMP-3                                S9(7)V99  75        5
03 FILLER                                DISPLAY                                X        80        1
IDMSRPTS 14.1                          -- SCHEMA RECORD DESCRIPTION LISTING ---          DATE      TIME      PAGE
RECDES                                DICTIONARY DEFAULT  OF NODE DEFAULT          02/13/01  142430      5
                                SCHEMA EMPSCHM  VERSION  100

REC SYNONYM NAME... DNTCLM                                DENTAL-CLAIM                                RLGTH=  948
REC SYNONYM VER.... 0100                                DLGTH=  936
```

```

LANGUAGE(S)..... FORTRAN
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH

02 DCCLDT                DISPLAY                1      8
03 DCCLYR                DISPLAY                9(4)    1      4
03 DCCLMO                DISPLAY                9(2)    5      2
03 DCCLDA                DISPLAY                9(2)    7      2
02 DCPNAM                DISPLAY                9      25
03 DCPFNM                DISPLAY                X(10)   9      10
03 DCPPLNM               DISPLAY                X(15)  19      15
02 DCPBDT                DISPLAY                34      8
03 DCPBYR                DISPLAY                9(4)   34      4
03 DCPBMO                DISPLAY                9(2)   38      2
03 DCPBDA                DISPLAY                9(2)   40      2
02 DCPSEX                DISPLAY                X      42      1
02 DCRLEMP               DISPLAY                X(10)  43      10
02 DCDNNM                DISPLAY                53      25
03 DCDNFM                DISPLAY                X(10)  53      10
03 DCDNLN                DISPLAY                X(15)  63      15
02 DCDNAD                DISPLAY                78      46
03 DCDNST                DISPLAY                X(20)  78      20
03 DCDNCY                DISPLAY                X(15)  98      15
03 DCDNS                 DISPLAY                X(2)   113     2
03 DCDNZP                DISPLAY                115     9
04 DCDNZ5                DISPLAY                X(5)   115     5
04 DCDNZ4                DISPLAY                X(4)   120     4
02 DCLICN                DISPLAY                9(6)   124     6
02 DCNOPR                COMP                  9(2)   130     2
02 FILLER                DISPLAY                X      132     1
02 DCDNCH                DISPLAY                OCCURS 0 TO 10
                        DEPENDING ON ----- DCNOPR
03 DCTTHN                DISPLAY                9(2)    1      2
03 DCSVDT                DISPLAY                3      8
04 DCSVYR                DISPLAY                9(4)    3      4
04 DCSVMO                DISPLAY                9(2)    7      2
04 DCSVDA                DISPLAY                9(2)    9      2
03 DCPRCD                DISPLAY                9(4)   11      4
03 DCDSCS                DISPLAY                X(60)   15     60
03 DCFEE                 COMP-3        S9(7)V99   75      5
03 FILLER                DISPLAY                X      80      1

IDMSRPTS 14.1          -- SCHEMA RECORD DESCRIPTION LISTING ---      DATE      TIME  PAGE
RECDES                  DICTIONARY DEFAULT  OF NODE DEFAULT      02/13/01  142430  6
                        SCHEMA EMPSCHM  VERSION 100

RECORD NAME..... DEPARTMENT                RLGTH=   72
RECORD VERSION..... 0100                    DLGTH=   56
RECORD ID..... 0410                        KLGTH=   16
RECORD LENGTH..... FIXED                    DSTRT=   16
LOCATION MODE..... CALC USING  DEPT-ID-0410          DUPLICATES NOT ALLOWED
WITHIN..... ORG-DEMO-REGION  OFFSET              5 PGS FOR      45 PGS
DBKEY POSITIONS.... SET..... TYPE..... NEXT  PRIOR OWNER
                        CALC      MEMBER      1      2
                        DEPT-EMPLOYEE  OWNER      3      4

```



```
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....
STRT LGTH

02 DEPT-ID-0410          DISPLAY          9(4)          1      4
                        SET CONTROL ITEM FOR ----- CALC          ASC DUP

NOT ALLOWED
02 DEPT-NAME-0410        DISPLAY          X(45)          5     45
02 DEPT-HEAD-ID-0410     DISPLAY          9(4)          50     4
02 FILLER                DISPLAY          XXX           54     3

*****

REC SYNONYM NAME... DEPARTMT          DEPARTMENT          RLGTH=   72
REC SYNONYM VER.... 0100              DLGTH=   56
LANGUAGE(S)..... ASSEMBLER
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....  STRT LGTH

02 DEPTID                DISPLAY          9(4)          1      4
02 DEPTNAME              DISPLAY          X(45)          5     45
02 DEPTHID              DISPLAY          9(4)          50     4
02 FILLER                DISPLAY          XXX           54     3

*****

REC SYNONYM NAME... DEPT              DEPARTMENT          RLGTH=   72
REC SYNONYM VER.... 0100              DLGTH=   56
LANGUAGE(S)..... FORTRAN
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....  STRT LGTH

02 DPID                  DISPLAY          9(4)          1      4
02 DPNAME                DISPLAY          X(45)          5     45
02 DPHID                DISPLAY          9(4)          50     4
02 FILLER                DISPLAY          XXX           54     3
IDMSRPTS 14.1            -- SCHEMA RECORD DESCRIPTION LISTING ---      DATE      TIME  PAGE
RECDES                   DICTIONARY DEFAULT OF NODE DEFAULT      02/13/01 142430   7
                        SCHEMA EMPSCHM VERSION 100

RECORD NAME..... EMPLOYEE          RLGTH=  192
RECORD VERSION.... 0100              DLGTH=  120
RECORD ID..... 0415                KLGH=   72
RECORD LENGTH..... FIXED            DSTRT=  72
LOCATION MODE..... CALC USING EMP-ID-0415          DUPLICATES NOT ALLOWED
WITHIN..... EMP-DEMO-REGION          OFFSET          5 PGS FOR          95 PGS
DBKEY POSITIONS.... SET..... TYPE..... NEXT PRIOR OWNER
                        CALC          MEMBER          1      2
                        DEPT-EMPLOYEE MEMBER          3      4      5
                        EMP-NAME-NDX  INDEX MEMBER    6
                        OFFICE-EMPLOYEE INDEX MEMBER  7      8
                        EMP-COVERAGE  OWNER          9     10
                        EMP-EMPOSITION OWNER         11     12
                        EMP-EXPERTISE  OWNER         13     14
                        MANAGES        OWNER         15     16
                        REPORTS-TO     OWNER         17     18
```

DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....	STRT	LGTH
02 EMP-ID-0415		DISPLAY		9(4)	1	4
		SET CONTROL ITEM FOR -----		CALC	ASC DUP	
NOT ALLOWED						
02 EMP-NAME-0415		DISPLAY			5	25
03 EMP-FIRST-NAME-0415		DISPLAY		X(10)	5	10
		SET CONTROL ITEM FOR -----		DEPT-EMPLOYEE	ASC DUP LAST	
		SET CONTROL ITEM FOR -----		EMP-NAME-NDX	ASC DUP LAST	
		SET CONTROL ITEM FOR -----		OFFICE-EMPLOYEE	ASC DUP LAST	
03 EMP-LAST-NAME-0415		DISPLAY		X(15)	15	15
		SET CONTROL ITEM FOR -----		DEPT-EMPLOYEE	ASC DUP LAST	
		SET CONTROL ITEM FOR -----		EMP-NAME-NDX	ASC DUP LAST	
		SET CONTROL ITEM FOR -----		OFFICE-EMPLOYEE	ASC DUP LAST	
02 EMP-ADDRESS-0415		DISPLAY			30	46
03 EMP-STREET-0415		DISPLAY		X(20)	30	20
03 EMP-CITY-0415		DISPLAY		X(15)	50	15
03 EMP-STATE-0415		DISPLAY		X(2)	65	2
03 EMP-ZIP-0415		DISPLAY			67	9
04 EMP-ZIP-FIRST-FIVE-0415		DISPLAY		X(5)	67	5
04 EMP-ZIP-LAST-FOUR-0415		DISPLAY		X(4)	72	4
02 EMP-PHONE-0415		DISPLAY		9(10)	76	10
02 STATUS-0415		DISPLAY		X(2)	86	2
88 ACTIVE-0415		COND	'01'		86	
88 ST-DISABIL-0415		COND	'02'		86	
88 LT-DISABIL-0415		COND	'03'		86	
88 LEAVE-OF-ABSENCE-0415		COND	'04'		86	
88 TERMINATED-0415		COND	'05'		86	
02 SS-NUMBER-0415		DISPLAY		9(9)	88	9
02 START-DATE-0415		DISPLAY			97	8
03 START-YEAR-0415		DISPLAY		9(4)	97	4
03 START-MONTH-0415		DISPLAY		9(2)	101	2
03 START-DAY-0415		DISPLAY		9(2)	103	2
02 TERMINATION-DATE-0415		DISPLAY			105	8
03 TERMINATION-YEAR-0415		DISPLAY		9(4)	105	4
03 TERMINATION-MONTH-0415		DISPLAY		9(2)	109	2
IDMSRPTS 14.1	--	SCHEMA RECORD DESCRIPTION LISTING ---		DATE	TIME	PAGE
RECDES		DICTIONARY DEFAULT OF NODE DEFAULT		02/13/01	142430	8
		SCHEMA EMPSCHM VERSION 100				
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH						
03 TERMINATION-DAY-0415		DISPLAY		9(2)	111	2
02 BIRTH-DATE-0415		DISPLAY			113	8
03 BIRTH-YEAR-0415		DISPLAY		9(4)	113	4
03 BIRTH-MONTH-0415		DISPLAY		9(2)	117	2
03 BIRTH-DAY-0415		DISPLAY		9(2)	119	2

REC SYNONYM NAME...	EMPLOYEE		EMPLOYEE		RLGTH=	192
REC SYNONYM VER....	0100				DLGTH=	120
LANGUAGE(S).....	ASSEMBLER					

DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....	STRT	LGTH
02 EMPID		DISPLAY		9(4)	1	4
02 EMPNAME		DISPLAY			5	25
03 EMPFNAME		DISPLAY		X(10)	5	10
03 EMPLNAME		DISPLAY		X(15)	15	15
02 EMPADDR		DISPLAY			30	46
03 EMPSTRET		DISPLAY		X(20)	30	20
03 EMPCITY		DISPLAY		X(15)	50	15
03 EMPSTATE		DISPLAY		X(2)	65	2
03 EMPZIP		DISPLAY			67	9
04 EMPZIPF5		DISPLAY		X(5)	67	5
04 EMPZIPL4		DISPLAY		X(4)	72	4
02 EMPPHONE		DISPLAY		9(10)	76	10
02 EMPSTATU		DISPLAY		X(2)	86	2
88 ACTIVE		COND	'01'		86	
88 STDSBL		COND	'02'		86	
88 LTDSBL		COND	'03'		86	
88 LVOFAB		COND	'04'		86	
88 TRMINATD		COND	'05'		86	
02 EMPSSNUM		DISPLAY		9(9)	88	9
02 EMPSTDT		DISPLAY			97	8
03 EMPSTYR		DISPLAY		9(4)	97	4
03 EMPSTMO		DISPLAY		9(2)	101	2
03 EMPSTDA		DISPLAY		9(2)	103	2
02 EMPTRMDT		DISPLAY			105	8
03 EMPTRMYR		DISPLAY		9(4)	105	4
03 EMPTRMMO		DISPLAY		9(2)	109	2
03 EMPTRMDA		DISPLAY		9(2)	111	2
02 EMPBIRDY		DISPLAY			113	8
03 EMPBIRYR		DISPLAY		9(4)	113	4
03 EMPBIRMO		DISPLAY		9(2)	117	2
03 EMPBIRDA		DISPLAY		9(2)	119	2
IDMSRPTS 14.1	-- SCHEMA RECORD DESCRIPTION LISTING --				DATE	TIME
RECDES	DICTIONARY DEFAULT OF NODE DEFAULT				02/13/01	142430
	SCHEMA EMPSCHM VERSION 100					9
REC SYNONYM NAME... EMPLOY		EMPLOYEE			RLGTH=	192
REC SYNONYM VER.... 0100					DLGTH=	120
LANGUAGE(S)..... FORTRAN						
DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....	STRT	LGTH
02 EMPID		DISPLAY		9(4)	1	4
02 EMPNAM		DISPLAY			5	25
03 EMPFNM		DISPLAY		X(10)	5	10
03 EMPLNM		DISPLAY		X(15)	15	15
02 EMPADR		DISPLAY			30	46
03 EMPST		DISPLAY		X(20)	30	20
03 EMPCTY		DISPLAY		X(15)	50	15
03 EMPSTE		DISPLAY		X(2)	65	2
03 EMPZIP		DISPLAY			67	9
04 EMPZF5		DISPLAY		X(5)	67	5

04 EMPZL4	DISPLAY	X(4)	72	4
02 EMPPHN	DISPLAY	9(10)	76	10
02 EMPSTA	DISPLAY	X(2)	86	2
88 ACTIVE	COND	'01'	86	
88 STDSBL	COND	'02'	86	
88 LTDSBL	COND	'03'	86	
88 LVOFAB	COND	'04'	86	
88 TRMNTD	COND	'05'	86	
02 EMPSS	DISPLAY	9(9)	88	9
02 EMPSDT	DISPLAY		97	8
03 EMPSYR	DISPLAY	9(4)	97	4
03 EMPSMO	DISPLAY	9(2)	101	2
03 EMPSDA	DISPLAY	9(2)	103	2
02 EMPTDT	DISPLAY		105	8
03 EMPTYR	DISPLAY	9(4)	105	4
03 EMPTMO	DISPLAY	9(2)	109	2
03 EMPTDA	DISPLAY	9(2)	111	2
02 EMPBDT	DISPLAY		113	8
03 EMPBYR	DISPLAY	9(4)	113	4
03 EMPBMO	DISPLAY	9(2)	117	2
03 EMPBDA	DISPLAY	9(2)	119	2
IDMSRPTS 14.1	-- SCHEMA RECORD DESCRIPTION LISTING --			
RECDES	DICTIONARY DEFAULT	OF NODE DEFAULT	02/13/01	142430
	SCHEMA EMPSCHM VERSION 100			10
RECORD NAME.....	EMPOSITION		RLGTH=	56
RECORD VERSION.....	0100		DLGTH=	32
RECORD ID.....	0420		KLGTH=	24
RECORD LENGTH.....	FIXED		DSTRT=	24
LOCATION MODE.....	VIA SET	EMP-EMPOSITION	DISPLACEMENT 0000	PAGES
WITHIN.....	EMP-DEMO-REGION	OFFSET	5 PGS FOR	95 PGS
DBKEY POSITIONS....	SET.....	TYPE.....	NEXT	PRIOR OWNER
	EMP-EMPOSITION	MEMBER	1	2 3
	JOB-EMPOSITION	MEMBER	4	5 6
DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....
				STRT LGTH
02 START-DATE-0420	DISPLAY			1 8
03 START-YEAR-0420	DISPLAY	9(4)		1 4
03 START-MONTH-0420	DISPLAY	9(2)		5 2
03 START-DAY-0420	DISPLAY	9(2)		7 2
02 FINISH-DATE-0420	DISPLAY			9 8
03 FINISH-YEAR-0420	DISPLAY	9(4)		9 4
03 FINISH-MONTH-0420	DISPLAY	9(2)		13 2
03 FINISH-DAY-0420	DISPLAY	9(2)		15 2
02 SALARY-GRADE-0420	DISPLAY	9(2)		17 2
02 SALARY-AMOUNT-0420	COMP-3	S9(7)V99		19 5
02 BONUS-PERCENT-0420	COMP-3	SV999		24 2
02 COMMISSION-PERCENT-0420	COMP-3	SV999		26 2
02 OVERTIME-RATE-0420	COMP-3	S9V99		28 2
02 FILLER	DISPLAY	XXX		30 3

```
REC SYNONYM NAME... EMPOSITN                      EMPOSITION                      RLGTH= 56
REC SYNONYM VER.... 0100                          DLGTH= 32
LANGUAGE(S)..... ASSEMBLER
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH

02 EPSTRDTT                      DISPLAY                      1      8
03 EPSTRTYR                      DISPLAY                      9(4)    1      4
03 EPSTRTMO                      DISPLAY                      9(2)    5      2
03 EPSTRTDA                      DISPLAY                      9(2)    7      2
02 EPFINIDT                      DISPLAY                      9      8
03 EPFINIYR                      DISPLAY                      9(4)    9      4
03 EPFINIMO                      DISPLAY                      9(2)   13      2
03 EPFINIDA                      DISPLAY                      9(2)   15      2
02 EPSALGRD                      DISPLAY                      9(2)   17      2
02 EPSALAMT                      COMP-3                      S9(7)V99 19      5
02 EPBONPCT                      COMP-3                      SV999    24      2
02 EPCMPMPCT                     COMP-3                      SV999    26      2
02 EPOTRATE                      COMP-3                      S9V99    28      2
02 FILLER                        DISPLAY                      XXX      30      3

IDMSRPTS 14.1          -- SCHEMA RECORD DESCRIPTION LISTING ---          DATE      TIME      PAGE
RECDES                DICTIONARY DEFAULT OF NODE DEFAULT          02/13/01 142430 11
                      SCHEMA EMPSCHM VERSION 100

REC SYNONYM NAME... EMPOST                      EMPOSITION                      RLGTH= 56
REC SYNONYM VER.... 0100                          DLGTH= 32
LANGUAGE(S)..... FORTRAN
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH

02 EPSTDT                      DISPLAY                      1      8
03 EPSTYR                      DISPLAY                      9(4)    1      4
03 EPSTMO                      DISPLAY                      9(2)    5      2
03 EPSTDA                      DISPLAY                      9(2)    7      2
02 EPFNDT                      DISPLAY                      9      8
03 EPFNYR                      DISPLAY                      9(4)    9      4
03 EPFNMO                      DISPLAY                      9(2)   13      2
03 EPFNDA                      DISPLAY                      9(2)   15      2
02 EPSLGD                      DISPLAY                      9(2)   17      2
02 EPSLAM                      COMP-3                      S9(7)V99 19      5
02 EPBPCT                      COMP-3                      SV999    24      2
02 EPCPCT                      COMP-3                      SV999    26      2
02 EPOTRT                      COMP-3                      S9V99    28      2
02 FILLER                        DISPLAY                      XXX      30      3

IDMSRPTS 14.1          -- SCHEMA RECORD DESCRIPTION LISTING ---          DATE      TIME      PAGE
RECDES                DICTIONARY DEFAULT OF NODE DEFAULT          02/13/01 142430 12
                      SCHEMA EMPSCHM VERSION 100

RECORD NAME..... EXPERTISE                      RLGTH= 32
RECORD VERSION.... 0100                          DLGTH= 12
RECORD ID..... 0425                             KLGH= 20
RECORD LENGTH..... FIXED                         DSTRT= 20

LOCATION MODE..... VIA SET      EMP-EXPERTISE      DISPLACEMENT 0000 PAGES
WITHIN..... EMP-DEMO-REGION      OFFSET      5 PGS FOR      95 PGS
DBKEY POSITIONS.... SET..... TYPE..... NEXT      PRIOR OWNER
```

EMP-EXPERTISE	MEMBER	1	2	3			
SKILL-EXPERTISE	INDEX MEMBER	4		5			
DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....	STRT	LGTH	
02 SKILL-LEVEL-0425	DISPLAY			XX	1	2	
	SET CONTROL ITEM FOR -----			EMP-EXPERTISE	DSC DUP	FIRST	
	SET CONTROL ITEM FOR -----			SKILL-EXPERTISE	DSC DUP	FIRST	
88 EXPERT-0425	COND	'04'			1		
88 PROFICIENT-0425	COND	'03'			1		
88 COMPETENT-0425	COND	'02'			1		
88 ELEMENTARY-0425	COND	'01'			1		
02 EXPERTISE-DATE-0425	DISPLAY				3	8	
03 EXPERTISE-YEAR-0425	DISPLAY			9(4)	3	4	
03 EXPERTISE-MONTH-0425	DISPLAY			9(2)	7	2	
03 EXPERTISE-DAY-0425	DISPLAY			9(2)	9	2	
02 FILLER	DISPLAY			XX	11	2	

REC SYNONYM NAME... EXPRTISE		EXPERTISE			RLGTH=	32	
REC SYNONYM VER.... 0100					DLGTH=	12	
LANGUAGE(S).....	ASSEMBLER						
DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....	STRT	LGTH	
02 EXPSKLVL	DISPLAY			XX	1	2	
88 EXPERT	COND	'04'			1		
88 PROFICNT	COND	'03'			1		
88 COMPETNT	COND	'02'			1		
88 ELEMNTY	COND	'01'			1		
02 EXPDATE	DISPLAY				3	8	
03 EXPYEAR	DISPLAY			9(4)	3	4	
03 EXPMONTH	DISPLAY			9(2)	7	2	
03 EXPDAY	DISPLAY			9(2)	9	2	
02 FILLER	DISPLAY			XX	11	2	
IDMSRPTS 14.1	-- SCHEMA RECORD DESCRIPTION LISTING ---				DATE	TIME	PAGE
RECDES	DICTIONARY DEFAULT	OF NODE DEFAULT			02/13/01	142430	13
	SCHEMA EMPSCHM	VERSION 100					
REC SYNONYM NAME... EXPRTS		EXPERTISE			RLGTH=	32	
REC SYNONYM VER.... 0100					DLGTH=	12	
LANGUAGE(S).....	FORTRAN						
DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....	STRT	LGTH	
02 EXPSKL	DISPLAY			XX	1	2	
88 EXPERT	COND	'04'			1		
88 PRFCNT	COND	'03'			1		
88 CMPINT	COND	'02'			1		
88 ELMNTY	COND	'01'			1		
02 EXPDTE	DISPLAY				3	8	
03 EXPYR	DISPLAY			9(4)	3	4	
03 EXPMO	DISPLAY			9(2)	7	2	
03 EXPDAY	DISPLAY			9(2)	9	2	
02 FILLER	DISPLAY			XX	11	2	

```

IDMSRPTS 14.1          -- SCHEMA RECORD DESCRIPTION LISTING ---          DATE      TIME      PAGE
RECDDES                DICTIONARY DEFAULT  OF NODE DEFAULT          02/13/01  142430   14
                        SCHEMA EMPSCHM  VERSION 100

RECORD NAME..... HOSPITAL-CLAIM                                RLGTH=  308
RECORD VERSION..... 0100                                         DLGTH=  300
RECORD ID..... 0430                                           KLGTH=    8
RECORD LENGTH..... FIXED                                       DSTRT=    8

LOCATION MODE..... VIA SET      COVERAGE-CLAIMS      DISPLACEMENT 0000  PAGES
WITHIN..... INS-DEMO-REGION      OFFSET      5 PGS FOR      45 PGS
DBKEY POSITIONS.... SET..... TYPE..... NEXT  PRIOR OWNER
                        COVERAGE-CLAIMS  MEMBER      1      2
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....  STRT  LGTH

02 CLAIM-DATE-0430                DISPLAY                                1      8
03 CLAIM-YEAR-0430                DISPLAY                                1      4
03 CLAIM-MONTH-0430              9(4)                                5      2
03 CLAIM-DAY-0430                9(2)                                7      2
02 PATIENT-NAME-0430              DISPLAY                                9     25
03 PATIENT-FIRST-NAME-0430        DISPLAY                               X(10)    9     10
03 PATIENT-LAST-NAME-0430         DISPLAY                               X(15)   19     15
02 PATIENT-BIRTH-DATE-0430        DISPLAY                               34      8
03 PATIENT-BIRTH-YEAR-0430        DISPLAY                               9(4)    34      4
03 PATIENT-BIRTH-MONTH-0430       DISPLAY                               9(2)    38      2
03 PATIENT-BIRTH-DAY-0430         DISPLAY                               9(2)    40      2
02 PATIENT-SEX-0430               DISPLAY                               X        42      1
02 RELATION-TO-EMPLOYEE-0430       DISPLAY                               X(10)   43     10
02 HOSPITAL-NAME-0430              DISPLAY                               X(25)   53     25
02 HOSP-ADDRESS-0430              DISPLAY                               78     46
03 HOSP-STREET-0430               DISPLAY                               X(20)   78     20
03 HOSP-CITY-0430                 DISPLAY                               X(15)   98     15
03 HOSP-STATE-0430                DISPLAY                               X(2)   113      2
03 HOSP-ZIP-0430                  DISPLAY                               115      9
04 HOSP-ZIP-FIRST-FIVE-0430        DISPLAY                               X(5)   115      5
04 HOSP-ZIP-LAST-FOUR-0430         DISPLAY                               X(4)   120      4
02 ADMIT-DATE-0430                DISPLAY                               124      8
03 ADMIT-YEAR-0430                DISPLAY                               9(4)   124      4
03 ADMIT-MONTH-0430               DISPLAY                               9(2)   128      2
03 ADMIT-DAY-0430                 DISPLAY                               9(2)   130      2
02 DISCHARGE-DATE-0430             DISPLAY                               132      8
03 DISCHARGE-YEAR-0430             DISPLAY                               9(4)   132      4
03 DISCHARGE-MONTH-0430            DISPLAY                               9(2)   136      2
03 DISCHARGE-DAY-0430             DISPLAY                               9(2)   138      2
02 DIAGNOSIS-0430                 DISPLAY      OCCURS      2      X(60)   140    120
02 HOSPITAL-CHARGES-0430           DISPLAY                               260     41
03 ROOM-AND-BOARD-0430            DISPLAY                               260     26
04 WARD-0430                      DISPLAY                               260     13
05 WARD-DAYS-0430                  COMP-3                               S9(5)   260      3
05 WARD-RATE-0430                  COMP-3                               S9(7)V99 263      5
05 WARD-TOTAL-0430                 COMP-3                               S9(7)V99 268      5
04 SEMI-PRIVATE-0430              DISPLAY                               273     13
05 SEMI-DAYS-0430                  COMP-3                               S9(5)   273      3
05 SEMI-RATE-0430                  COMP-3                               S9(7)V99 276      5

```

05 SEMI-TOTAL-0430	COMP-3	S9(7)V99	281	5
03 OTHER-CHARGES-0430	DISPLAY		286	15
IDMSRPTS 14.1	-- SCHEMA RECORD DESCRIPTION LISTING ---	DATE	TIME	PAGE
RECDES	DICTIONARY DEFAULT OF NODE DEFAULT	02/13/01	142430	15
	SCHEMA EMPSCHM VERSION 100			

DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....	STRT	LGTH
04 DELIVERY-COST-0430	COMP-3		S9(7)V99		286	5
04 ANESTHESIA-COST-0430	COMP-3		S9(7)V99		291	5
04 LAB-COST-0430	COMP-3		S9(7)V99		296	5

REC SYNONYM NAME... HOSPCLM	HOSPITAL-CLAIM	RLGTH=	308
REC SYNONYM VER.... 0100		DLGTH=	300
LANGUAGE(S).....	ASSEMBLER		

DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....	STRT	LGTH
02 HCCLMDT	DISPLAY				1	8
03 HCCLMYR	DISPLAY			9(4)	1	4
03 HCCLMMO	DISPLAY			9(2)	5	2
03 HCCLMDAY	DISPLAY			9(2)	7	2
02 HCPTNAME	DISPLAY				9	25
03 HCPTFNAM	DISPLAY			X(10)	9	10
03 HCPTLNAM	DISPLAY			X(15)	19	15
02 HCPTBDAT	DISPLAY				34	8
03 HCPTBYR	DISPLAY			9(4)	34	4
03 HCPTBMO	DISPLAY			9(2)	38	2
03 HCPTBDA	DISPLAY			9(2)	40	2
02 HCPTSEX	DISPLAY			X	42	1
02 HCRELEMP	DISPLAY			X(10)	43	10
02 HCHSPNAM	DISPLAY			X(25)	53	25
02 HCHSPADD	DISPLAY				78	46
03 HCHSPSTR	DISPLAY			X(20)	78	20
03 HCHSPCTY	DISPLAY			X(15)	98	15
03 HCHSPSTA	DISPLAY			X(2)	113	2
03 HCHSPZIP	DISPLAY				115	9
04 HCHSPZF5	DISPLAY			X(5)	115	5
04 HCHSPZL4	DISPLAY			X(4)	120	4
02 HCADMTDT	DISPLAY				124	8
03 HCADMTYR	DISPLAY			9(4)	124	4
03 HCADMTMO	DISPLAY			9(2)	128	2
03 HCADMTDA	DISPLAY			9(2)	130	2
02 HCDS CGDT	DISPLAY				132	8
03 HCDS CGYR	DISPLAY			9(4)	132	4
03 HCDS CGMO	DISPLAY			9(2)	136	2
03 HCDS CGDA	DISPLAY			9(2)	138	2
02 HCDIAGN	DISPLAY	OCCURS	2	X(60)	140	120
02 HCHSPCHG	DISPLAY				260	41
03 HCRM BRD	DISPLAY				260	26
04 HCWARD	DISPLAY				260	13
05 HCWDDAYS	COMP-3			S9(5)	260	3

05 HCWDRATE	COMP-3	S9(7)V99	263	5
IDMSRPTS 14.1	-- SCHEMA RECORD DESCRIPTION LISTING ---	DATE	TIME	PAGE
RECDES	DICTIONARY DEFAULT OF NODE DEFAULT	02/13/01	142430	16
	SCHEMA EMPSCHM VERSION 100			

DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....

STRT LGTH

05 HCWDTOTL	COMP-3	S9(7)V99	268	5
04 HCSPRIV	DISPLAY		273	13
05 HCSDBAYS	COMP-3	S9(5)	273	3
05 HCSRATE	COMP-3	S9(7)V99	276	5
05 HCSOTAL	COMP-3	S9(7)V99	281	5
03 HCOTHCHG	DISPLAY		286	15
04 HCDELVCH	COMP-3	S9(7)V99	286	5
04 HCANSTHC	COMP-3	S9(7)V99	291	5
04 HCLABCST	COMP-3	S9(7)V99	296	5

REC SYNONYM NAME... HSPCLM	HOSPITAL-CLAIM	RLGTH=	308
REC SYNONYM VER.... 0100		DLGTH=	300

LANGUAGE(S)..... FORTRAN

DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....

STRT LGTH

02 HCCLDT	DISPLAY		1	8
03 HCCLYR	DISPLAY	9(4)	1	4
03 HCCLMO	DISPLAY	9(2)	5	2
03 HCCLDA	DISPLAY	9(2)	7	2
02 HCPTNM	DISPLAY		9	25
03 HCPTFN	DISPLAY	X(10)	9	10
03 HCPTLN	DISPLAY	X(15)	19	15
02 HCPBDT	DISPLAY		34	8
03 HCPBYR	DISPLAY	9(4)	34	4
03 HCPBMO	DISPLAY	9(2)	38	2
03 HCPBDA	DISPLAY	9(2)	40	2
02 HCPTSEX	DISPLAY	X	42	1
02 HCREMP	DISPLAY	X(10)	43	10
02 HCHNAM	DISPLAY	X(25)	53	25
02 HCHADD	DISPLAY		78	46
03 HCHSTR	DISPLAY	X(20)	78	20
03 HCHCTY	DISPLAY	X(15)	98	15
03 HCHSTA	DISPLAY	X(2)	113	2
03 HCHZIP	DISPLAY		115	9
04 HCHZF5	DISPLAY	X(5)	115	5
04 HCHZL4	DISPLAY	X(4)	120	4
02 HCADDT	DISPLAY		124	8
03 HCADYR	DISPLAY	9(4)	124	4
03 HCADMO	DISPLAY	9(2)	128	2
03 HCADDA	DISPLAY	9(2)	130	2
02 HCDSDT	DISPLAY		132	8
03 HCDSYR	DISPLAY	9(4)	132	4
03 HCDSMO	DISPLAY	9(2)	136	2

```

03 HCDSDA                      DISPLAY                      9(2)          138      2
IDMSRPTS 14.1                  -- SCHEMA RECORD DESCRIPTION LISTING ---      DATE      TIME  PAGE
RECDES                          DICTIONARY DEFAULT  OF NODE DEFAULT      02/13/01  142430  17
                                SCHEMA EMPSCHM  VERSION 100

DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....  STRT  LGTH

02 HCDIAG                      DISPLAY      OCCURS      2      X(60)          140  120
02 HCHCHG                      DISPLAY                                260   41
03 HCRMBD                      DISPLAY                                260   26
04 HCWARD                      DISPLAY                                260   13
05 HCWDYS                      COMP-3              S9(5)          260    3
05 HCWRTE                      COMP-3              S9(7)V99       263    5
05 HCWTOT                      COMP-3              S9(7)V99       268    5
04 HCSPRV                      DISPLAY                                273   13
05 HCSDYS                      COMP-3              S9(5)          273    3
05 HCSRTE                      COMP-3              S9(7)V99       276    5
05 HCSTOT                      COMP-3              S9(7)V99       281    5
03 HCOCHG                      DISPLAY                                286   15
04 HCDLVC                      COMP-3              S9(7)V99       286    5
04 HCANSC                      COMP-3              S9(7)V99       291    5
04 HCLABC                      COMP-3              S9(7)V99       296    5
IDMSRPTS 14.1                  -- SCHEMA RECORD DESCRIPTION LISTING ---      DATE      TIME  PAGE
RECDES                          DICTIONARY DEFAULT  OF NODE DEFAULT      02/13/01  142430  18
                                SCHEMA EMPSCHM  VERSION 100

RECORD NAME..... INSURANCE-PLAN                                RLGTH= 140
RECORD VERSION..... 0100                                DLGTH= 132
RECORD ID..... 0435                                KLGTH= 8
RECORD LENGTH..... FIXED                                DSTRT= 8
LOCATION MODE..... CALC USING  INS-PLAN-CODE-0435      DUPLICATES NOT ALLOWED
WITHIN..... INS-DEMO-REGION  OFFSET      1 PGS FOR      4 PGS
DBKEY POSITIONS.... SET..... TYPE..... NEXT  PRIOR OWNER
                        CALC      MEMBER      1      2
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....  STRT  LGTH

02 INS-PLAN-CODE-0435          DISPLAY                      X(3)          1      3
                                SET CONTROL ITEM FOR -----  CALC      ASC DUP NOT ALLOWED
88 GROUP-LIFE-0435            COND      '001'          1
88 HMO-0435                   COND      '002'          1
88 GROUP-HEALTH-0435          COND      '003'          1
88 GROUP-DENTAL-0435          COND      '004'          1
02 INS-CO-NAME-0435            DISPLAY                      X(45)          4   45
02 INS-CO-ADDRESS-0435        DISPLAY                                49   46
03 INS-CO-STREET-0435          DISPLAY                      X(20)          49   20
03 INS-CO-CITY-0435            DISPLAY                      X(15)          69   15
03 INS-CO-STATE-0435          DISPLAY                      X(2)           84    2
03 INS-CO-ZIP-0435             DISPLAY                                86    9
04 INS-CO-ZIP-FIRST-FIVE-0435  DISPLAY                      X(5)           86    5
04 INS-CO-ZIP-LAST-FOUR-0435  DISPLAY                      X(4)           91    4
02 INS-CO-PHONE-0435           DISPLAY                      9(10)          95   10
02 GROUP-NUMBER-0435           DISPLAY                      9(6)          105    6
02 PLAN-DESCRIPTION-0435      DISPLAY                                111   20

```

03 DEDUCT-0435	COMP-3	S9(7)V99	111	5
03 MAXIMUM-LIFE-COST-0435	COMP-3	S9(7)V99	116	5
03 FAMILY-COST-0435	COMP-3	S9(7)V99	121	5
03 DEP-COST-0435	COMP-3	S9(7)V99	126	5
02 FILLER	DISPLAY	XX	131	2

REC SYNONYM NAME...	INSPLAN	INSURANCE-PLAN	RLGTH=	140
REC SYNONYM VER....	0100		DLGTH=	132
LANGUAGE(S).....	ASSEMBLER			
DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....
			STRT	LGTH

02 INPCODE	DISPLAY		X(3)	1	3
88 GROUPLIF	COND	'001'		1	
88 HMO	COND	'002'		1	
88 GRPHLTH	COND	'003'		1	
88 GROUPDNT	COND	'004'		1	
02 INPCNAME	DISPLAY		X(45)	4	45
02 INPCADDR	DISPLAY			49	46
03 INPCSTRT	DISPLAY		X(20)	49	20
IDMSRPTS 14.1	-- SCHEMA RECORD DESCRIPTION LISTING --		DATE	TIME	PAGE
RECDES	DICTIONARY DEFAULT	OF NODE DEFAULT	02/13/01	142430	19
	SCHEMA EMPSCHM	VERSION 100			

DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....	STRT	LGTH
----------------	----------------	------------	------------	--------------	------	------

03 INPCCITY	DISPLAY		X(15)	69	15
03 INPCSTAT	DISPLAY		X(2)	84	2
03 INPCZIP	DISPLAY			86	9
04 INPCZPF5	DISPLAY		X(5)	86	5
04 INPCZPL4	DISPLAY		X(4)	91	4
02 INPCPHON	DISPLAY		9(10)	95	10
02 INPGRPNO	DISPLAY		9(6)	105	6
02 INPDESCR	DISPLAY			111	20
03 INPDEDCT	COMP-3		S9(7)V99	111	5
03 INPMXLIF	COMP-3		S9(7)V99	116	5
03 INPFAMCS	COMP-3		S9(7)V99	121	5
03 INPDEPCS	COMP-3		S9(7)V99	126	5
02 FILLER	DISPLAY		XX	131	2

REC SYNONYM NAME...	INSPLN	INSURANCE-PLAN	RLGTH=	140
REC SYNONYM VER....	0100		DLGTH=	132
LANGUAGE(S).....	FORTTRAN			
DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....
			STRT	LGTH

02 IPCODE	DISPLAY		X(3)	1	3
88 GRPLIF	COND	'001'		1	
88 HMO	COND	'002'		1	
88 GRPHTH	COND	'003'		1	
88 GRPDNT	COND	'004'		1	

02 IPCNAM	DISPLAY	X(45)	4	45
02 IPCADR	DISPLAY		49	46
03 IPCSTR	DISPLAY	X(20)	49	20
03 IPCCTY	DISPLAY	X(15)	69	15
03 IPCSTA	DISPLAY	X(2)	84	2
03 IPCZIP	DISPLAY		86	9
04 IPCZF5	DISPLAY	X(5)	86	5
04 IPCZL4	DISPLAY	X(4)	91	4
02 IPCPHN	DISPLAY	9(10)	95	10
02 IPGRPN	DISPLAY	9(6)	105	6
02 IPDESC	DISPLAY		111	20
03 IPDDCT	COMP-3	S9(7)V99	111	5
03 IPMXLF	COMP-3	S9(7)V99	116	5
03 IPFMCS	COMP-3	S9(7)V99	121	5
03 IPDPSC	COMP-3	S9(7)V99	126	5
02 FILLER	DISPLAY	XX	131	2
IDMSRPTS 14.1	-- SCHEMA RECORD DESCRIPTION LISTING ---			
RECDES	DICTIONARY DEFAULT	OF NODE DEFAULT	DATE	TIME PAGE
			02/13/01	142430 20
	SCHEMA EMPSCHM VERSION 100			
RECORD NAME..... JOB			RLGTH=	324
RECORD VERSION..... 0100			DLGTH=	300
RECORD ID..... 0440			KLGTH=	24
RECORD LENGTH..... FIXED (INTERNALLY VARIABLE)			DSTRT=	28
MINIMUM ROOT..... 24 CHARACTERS				
MINIMUM FRAGMENT... 296 CHARACTERS				
LOCATION MODE..... CALC USING JOB-ID-0440		DUPLICATES NOT ALLOWED		
WITHIN..... ORG-DEMO-REGION OFFSET	5 PGS FOR	45 PGS		
CALL PROCEDURES.... NAME.... WHEN.. FUNCTION				
	IDMSCOMP BEFORE STORE			
	IDMSCOMP BEFORE MODIFY			
	IDMSDCOM AFTER GET			
DBKEY POSITIONS.... SET..... TYPE..... NEXT PRIOR OWNER				
	CALC MEMBER	1 2		
	JOB-TITLE-NDX INDEX MEMBER	3		
	JOB-EMPOSITION OWNER	4 5		
	(FRAGMENT CHAIN) INTRNL	6		
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....			STRT LGTH	
02 JOB-ID-0440	DISPLAY	9(4)	1	4
	SET CONTROL ITEM FOR -----	CALC	ASC DUP NOT ALLOWED	
02 TITLE-0440	DISPLAY	X(20)	5	20
	SET CONTROL ITEM FOR -----	JOB-TITLE-NDX	ASC DUP NOT ALLOWED	
02 DESCRIPTION-0440	DISPLAY		25	120
03 DESCRIPTION-LINE-0440	DISPLAY OCCURS	2 X(60)	25	120
02 REQUIREMENTS-0440	DISPLAY		145	120
03 REQUIREMENT-LINE-0440	DISPLAY OCCURS	2 X(60)	145	120
02 MINIMUM-SALARY-0440	DISPLAY	S9(6)V99	265	8
02 MAXIMUM-SALARY-0440	DISPLAY	S9(6)V99	273	8
02 SALARY-GRADES-0440	DISPLAY OCCURS	4 9(2)	281	8
02 NUMBER-OF-POSITIONS-0440	DISPLAY	9(3)	289	3
02 NUMBER-OPEN-0440	DISPLAY	9(3)	292	3
02 FILLER	DISPLAY	XX	295	2

```
*****

REC SYNONYM NAME... JOBA                                JOB                                RLGTH= 324
REC SYNONYM VER.... 0100                                DLGTH= 300
LANGUAGE(S)..... ASSEMBLER
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH

02 JOBID                                DISPLAY                                9(4)                                1      4
02 JOBTITLE                            DISPLAY                                X(20)                               5     20
02 JOBDESCR                            DISPLAY                                25     120
03 JOBDSCLN                            DISPLAY OCCURS 2 X(60)                25     120
02 JOBRQMNT                            DISPLAY                                145    120
03 JOBREQLN                            DISPLAY OCCURS 2 X(60)                145    120
02 JOBMNSAL                            DISPLAY                                S9(6)V99                            265     8
IDMSRPTS 14.1                          -- SCHEMA RECORD DESCRIPTION LISTING --- DATE TIME PAGE
RECDES                                DICTIONARY DEFAULT OF NODE DEFAULT 02/13/01 142430 21
                                SCHEMA EMPSCHM VERSION 100

DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH

02 JOBMXSAL                            DISPLAY                                S9(6)V99                            273     8
02 JOBSALGR                            DISPLAY OCCURS 4 9(2)                281     8
02 JOBNMPOS                            DISPLAY                                9(3)                                289     3
02 JOBNMOPN                            DISPLAY                                9(3)                                292     3
02 FILLER                              DISPLAY                                XX                                  295     2

*****

REC SYNONYM NAME... JOBF                                JOB                                RLGTH= 324
REC SYNONYM VER.... 0100                                DLGTH= 300
LANGUAGE(S)..... FORTRAN
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH

02 JOBID                                DISPLAY                                9(4)                                1      4
02 JTITLE                              DISPLAY                                X(20)                               5     20
02 JDESCR                              DISPLAY                                25     120
03 JDSCLN                              DISPLAY OCCURS 2 X(60)                25     120
02 JRQMNT                              DISPLAY                                145    120
03 JREQLN                              DISPLAY OCCURS 2 X(60)                145    120
02 JMNSAL                              DISPLAY                                S9(6)V99                            265     8
02 JMXSAL                              DISPLAY                                S9(6)V99                            273     8
02 JSALGR                              DISPLAY OCCURS 4 9(2)                281     8
02 JNMPOS                              DISPLAY                                9(3)                                289     3
02 JNMOPN                              DISPLAY                                9(3)                                292     3
02 FILLER                              DISPLAY                                XX                                  295     2
IDMSRPTS 14.1                          -- SCHEMA RECORD DESCRIPTION LISTING --- DATE TIME PAGE
RECDES                                DICTIONARY DEFAULT OF NODE DEFAULT 02/13/01 142430 22
                                SCHEMA EMPSCHM VERSION 100

RECORD NAME..... NON-HOSP-CLAIM                                RLGTH= 1068
RECORD VERSION.... 0100                                DLGTH= 1056
RECORD ID..... 0445                                KLGTH= 12
```

```

RECORD LENGTH..... VARIABLE                                DSTRT= 16
MINIMUM ROOT..... 248 CHARACTERS
MINIMUM FRAGMENT... 1052 CHARACTERS
LOCATION MODE..... VIA SET      COVERAGE-CLAIMS      DISPLACEMENT 0000 PAGES
WITHIN..... INS-DEMO-REGION      OFFSET      5 PGS FOR      45 PGS
DBKEY POSITIONS.... SET..... TYPE..... NEXT PRIOR OWNER
                        COVERAGE-CLAIMS MEMBER      1      2
                        (FRAGMENT CHAIN) INTRNL      3
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH

02 CLAIM-DATE-0445      DISPLAY      1      8
03 CLAIM-YEAR-0445      DISPLAY      9(4)      1      4
03 CLAIM-MONTH-0445     DISPLAY      9(2)      5      2
03 CLAIM-DAY-0445       DISPLAY      9(2)      7      2
02 PATIENT-NAME-0445     DISPLAY      9      25
03 PATIENT-FIRST-NAME-0445 DISPLAY      X(10)      9      10
03 PATIENT-LAST-NAME-0445 DISPLAY      X(15)     19      15
02 PATIENT-BIRTH-DATE-0445 DISPLAY      34      8
03 PATIENT-BIRTH-YEAR-0445 DISPLAY      9(4)     34      4
03 PATIENT-BIRTH-MONTH-0445 DISPLAY      9(2)     38      2
03 PATIENT-BIRTH-DAY-0445 DISPLAY      9(2)     40      2
02 PATIENT-SEX-0445      DISPLAY      X      42      1
02 RELATION-TO-EMPLOYEE-0445 DISPLAY      X(10)     43      10
02 PHYSICIAN-NAME-0445   DISPLAY      53      25
03 PHYSICIAN-FIRST-NAME-0445 DISPLAY      X(10)     53      10
03 PHYSICIAN-LAST-NAME-0445 DISPLAY      X(15)     63      15
02 PHYSICIAN-ADDRESS-0445 DISPLAY      78      46
03 PHYSICIAN-STREET-0445 DISPLAY      X(20)     78      20
03 PHYSICIAN-CITY-0445   DISPLAY      X(15)     98      15
03 PHYSICIAN-STATE-0445   DISPLAY      X(2)     113      2
03 PHYSICIAN-ZIP-0445     DISPLAY      115      9
04 PHYSICIAN-ZIP-FIRST-FIVE-0445 DISPLAY      X(5)     115      5
04 PHYSICIAN-ZIP-LAST-FOUR-0445 DISPLAY      X(4)     120      4
02 PHYSICIAN-ID-0445      DISPLAY      9(6)     124      6
02 DIAGNOSIS-0445         DISPLAY      OCCURS      2      X(60)     130     120
02 NUMBER-OF-PROCEDURES-0445 COMP      9(2)     250      2
02 FILLER                 DISPLAY      X      252      1
02 PHYSICIAN-CHARGES-0445 DISPLAY      OCCURS 0 TO 10      253     800
                        DEPENDING ON ----- NUMBER-OF-PROCEDURES-0445
03 SERVICE-DATE-0445      DISPLAY      1      8
04 SERVICE-YEAR-0445      DISPLAY      9(4)      1      4
04 SERVICE-MONTH-0445     DISPLAY      9(2)      5      2
04 SERVICE-DAY-0445       DISPLAY      9(2)      7      2
03 PROCEDURE-CODE-0445     DISPLAY      9(4)      9      4
03 DESCRIPTION-OF-SERVICE-0445 DISPLAY      X(60)     13      60
03 FEE-0445               COMP-3      S9(7)V99     73      5
03 FILLER                 DISPLAY      XXX      78      3
IDMSRPTS 14.1      -- SCHEMA RECORD DESCRIPTION LISTING ---      DATE      TIME      PAGE
RECDES      DICTIONARY DEFAULT OF NODE DEFAULT      02/13/01 142430 23
                        SCHEMA EMPSCHM VERSION 100

REC SYNONYM NAME... NONHSPCL      NON-HOSP-CLAIM      RLGTH= 1068
REC SYNONYM VER.... 0100      DLGTH= 1056

```

LANGUAGE(S)..... ASSEMBLER									
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....									
STRT LGTH									
02	NHCLMDT		DISPLAY					1	8
03	NHCLMYR		DISPLAY			9(4)		1	4
03	NHCLMMO		DISPLAY			9(2)		5	2
03	NHCLMDAY		DISPLAY			9(2)		7	2
02	NHPTNAME		DISPLAY					9	25
03	NHPTFNAM		DISPLAY			X(10)		9	10
03	NHPTLNAM		DISPLAY			X(15)		19	15
02	NHPTBDAT		DISPLAY					34	8
03	NHPTBYR		DISPLAY			9(4)		34	4
03	NHPTBMO		DISPLAY			9(2)		38	2
03	NHPTBDA		DISPLAY			9(2)		40	2
02	NHPTSEX		DISPLAY			X		42	1
02	NHRELEMP		DISPLAY			X(10)		43	10
02	NHPHYNAM		DISPLAY					53	25
03	NHPHYFNM		DISPLAY			X(10)		53	10
03	NHPHYLMN		DISPLAY			X(15)		63	15
02	NHPHYADD		DISPLAY					78	46
03	NHPHYSTR		DISPLAY			X(20)		78	20
03	NHPHYCTY		DISPLAY			X(15)		98	15
03	NHPHYSTA		DISPLAY			X(2)		113	2
03	NHPHYZIP		DISPLAY					115	9
04	NHPHYZ5		DISPLAY			X(5)		115	5
04	NHPHYZ4		DISPLAY			X(4)		120	4
02	NHPHYSID		DISPLAY			9(6)		124	6
02	NHDIAGN		DISPLAY	OCCURS	2	X(60)		130	120
02	NHNOPROC		COMP			9(2)		250	2
02	FILLER		DISPLAY			X		252	1
02	NHPHYCHG		DISPLAY	OCCURS 0 TO	10			253	800
DEPENDING ON -----							NHNOPROC		
03	NHSERVDI		DISPLAY					1	8
04	NHSERVYR		DISPLAY			9(4)		1	4
04	NHSERVMO		DISPLAY			9(2)		5	2
04	NHSERVDA		DISPLAY			9(2)		7	2
03	NHPROCDD		DISPLAY			9(4)		9	4
03	NHDESCSV		DISPLAY			X(60)		13	60
03	NHFEE		COMP-3			S9(7)V99		73	5
03	FILLER		DISPLAY			XXX		78	3
IDMSRPTS 14.1 -- SCHEMA RECORD DESCRIPTION LISTING --- DATE TIME PAGE									
RECDES DICTIONARY DEFAULT OF NODE DEFAULT 02/13/01 142430 24									
SCHEMA EMPSCHM VERSION 100									
REC SYNONYM NAME... NHSPCL NON-HOSP-CLAIM RLGTH= 1068									
REC SYNONYM VER.... 0100 DLGTH= 1056									
LANGUAGE(S)..... FORTRAN									
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE.....									
STRT LGTH									
02	NHCLDT		DISPLAY					1	8
03	NHCLYR		DISPLAY			9(4)		1	4
03	NHCLMO		DISPLAY			9(2)		5	2

03 NHCLDA	DISPLAY	9(2)	7	2
02 NHPTNM	DISPLAY		9	25
03 NHPTFN	DISPLAY	X(10)	9	10
03 NHPTLN	DISPLAY	X(15)	19	15
02 NHPBDT	DISPLAY		34	8
03 NHPBYR	DISPLAY	9(4)	34	4
03 NHPBMO	DISPLAY	9(2)	38	2
03 NHPBDA	DISPLAY	9(2)	40	2
02 NHPSEX	DISPLAY	X	42	1
02 NHREMP	DISPLAY	X(10)	43	10
02 NHPHNM	DISPLAY		53	25
03 NHPHFN	DISPLAY	X(10)	53	10
03 NHPHLN	DISPLAY	X(15)	63	15
02 NHPHAD	DISPLAY		78	46
03 NHPHST	DISPLAY	X(20)	78	20
03 NHPHCY	DISPLAY	X(15)	98	15
03 NHPHSA	DISPLAY	X(2)	113	2
03 NHPHZP	DISPLAY		115	9
04 NHPHZ5	DISPLAY	X(5)	115	5
04 NHPHZ4	DISPLAY	X(4)	120	4
02 NHPHID	DISPLAY	9(6)	124	6
02 NHDIAG	DISPLAY	OCCURS 2	130	120
02 NHNPRC	COMP	9(2)	250	2
02 FILLER	DISPLAY	X	252	1
02 NHPHCH	DISPLAY	OCCURS 0 TO 10	253	800
	DEPENDING ON	-----	NHNPRC	
03 NHSDVT	DISPLAY		1	8
04 NHSVYR	DISPLAY	9(4)	1	4
04 NHSMVO	DISPLAY	9(2)	5	2
04 NHSDVA	DISPLAY	9(2)	7	2
03 NHPRCD	DISPLAY	9(4)	9	4
03 NHDSCS	DISPLAY	X(60)	13	60
03 NHFEE	COMP-3	S9(7)V99	73	5
03 FILLER	DISPLAY	XXX	78	3
IDMSRPTS 14.1	-- SCHEMA RECORD DESCRIPTION LISTING --	DATE	TIME	PAGE
RECDES	DICTIONARY DEFAULT OF NODE DEFAULT	02/13/01	142430	25
	SCHEMA EMPSCHM VERSION 100			
RECORD NAME..... OFFICE			RLGTH=	92
RECORD VERSION..... 0100			DLGTH=	76
RECORD ID..... 0450			KLGTH=	16
RECORD LENGTH..... FIXED			DSTRT=	16
LOCATION MODE..... CALC USING OFFICE-CODE-0450		DUPLICATES NOT ALLOWED		
WITHIN..... ORG-DEMO-REGION	OFFSET	5 PGS FOR	45 PGS	
DBKEY POSITIONS.... SET..... TYPE..... NEXT	PRIOR OWNER			
	CALC	MEMBER	1	2
	OFFICE-EMPLOYEE	INDEX OWNER	3	4
DATA ITEM..... REDEFINES..... USAGE..... VALUE.....	PICTURE.....	STRT	LGTH	
02 OFFICE-CODE-0450	DISPLAY	X(3)	1	3
	SET CONTROL ITEM FOR	-----	CALC	ASC DUP NOT ALLOWED
02 OFFICE-ADDRESS-0450	DISPLAY		4	46
03 OFFICE-STREET-0450	DISPLAY	X(20)	4	20

03 OFFICE-CITY-0450	DISPLAY			X(15)	24	15
03 OFFICE-STATE-0450	DISPLAY			X(2)	39	2
03 OFFICE-ZIP-0450	DISPLAY				41	9
04 OFFICE-ZIP-FIRST-FIVE-0450	DISPLAY			X(5)	41	5
04 OFFICE-ZIP-LAST-FOUR-0450	DISPLAY			X(4)	46	4
02 OFFICE-PHONE-0450	DISPLAY	OCCURS	3	9(7)	50	21
02 OFFICE-AREA-CODE-0450	DISPLAY			X(3)	71	3
02 SPEED-DIAL-0450	DISPLAY			X(3)	74	3

REC SYNONYM NAME...	OFFIC	OFFICE	RLGTH=	92
REC SYNONYM VER....	0100		DLGTH=	76
LANGUAGE(S).....	ASSEMBLER			
DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....
			STRT	LGTH

02 OFFCODE	DISPLAY			X(3)	1	3
02 OFFADDR	DISPLAY				4	46
03 OFFSTRT	DISPLAY			X(20)	4	20
03 OFFCITY	DISPLAY			X(15)	24	15
03 OFFSTATE	DISPLAY			X(2)	39	2
03 OFFZIP	DISPLAY				41	9
04 OFFZIPF5	DISPLAY			X(5)	41	5
04 OFFZIPL4	DISPLAY			X(4)	46	4
02 OFFPHONE	DISPLAY	OCCURS	3	9(7)	50	21
02 OFFAREA	DISPLAY			X(3)	71	3
02 OFFSPEED	DISPLAY			X(3)	74	3

IDMSRPTS 14.1	-- SCHEMA RECORD DESCRIPTION LISTING ---	DATE	TIME	PAGE
RECDES	DICTIONARY DEFAULT OF NODE DEFAULT	02/13/01	142430	26
	SCHEMA EMPSCHM VERSION 100			

REC SYNONYM NAME...	OFFCE	OFFICE	RLGTH=	92
REC SYNONYM VER....	0100		DLGTH=	76
LANGUAGE(S).....	FORTRAN			
DATA ITEM.....	REDEFINES.....	USAGE.....	VALUE.....	PICTURE.....
			STRT	LGTH

02 OFCODE	DISPLAY			X(3)	1	3
02 OFADDR	DISPLAY				4	46
03 OFSTRT	DISPLAY			X(20)	4	20
03 OFCITY	DISPLAY			X(15)	24	15
03 OFSTAT	DISPLAY			X(2)	39	2
03 OFZIP	DISPLAY				41	9
04 OFZPF5	DISPLAY			X(5)	41	5
04 OFZPL4	DISPLAY			X(4)	46	4
02 OFPHON	DISPLAY	OCCURS	3	9(7)	50	21
02 OFAREA	DISPLAY			X(3)	71	3
02 OFSPED	DISPLAY			X(3)	74	3

IDMSRPTS 14.1	-- SCHEMA RECORD DESCRIPTION LISTING ---	DATE	TIME	PAGE
RECDES	DICTIONARY DEFAULT OF NODE DEFAULT	02/13/01	142430	27
	SCHEMA EMPSCHM VERSION 100			

RECORD NAME.....	SKILL	RLGTH=	96
RECORD VERSION.....	0100	DLGTH=	76

```

RECORD ID..... 0455                                KLGT= 20
RECORD LENGTH..... FIXED                             DSTRT= 20
LOCATION MODE..... CALC USING SKILL-ID-0455             DUPLICATES NOT ALLOWED
WITHIN..... ORG-DEMO-REGION   OFFSET               5 PGS FOR      45 PGS
DBKEY POSITIONS... SET..... TYPE..... NEXT   PRIOR OWNER
                  CALC          MEMBER          1      2
                  SKILL-NAME-NDX INDEX MEMBER    3
                  SKILL-EXPERTISE INDEX OWNER    4      5
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH

02 SKILL-ID-0455          DISPLAY                  9(4)          1      4
                  SET CONTROL ITEM FOR ----- CALC      ASC DUP NOT ALLOWED
02 SKILL-NAME-0455        DISPLAY                  X(12)          5     12
                  SET CONTROL ITEM FOR ----- SKILL-NAME-NDX ASC DUP NOT ALLOWED
02 SKILL-DESCRIPTION-0455 DISPLAY                  X(60)         17     60

*****

REC SYNONYM NAME... SKILLA                          SKILL                      RLGT= 96
REC SYNONYM VER.... 0100                             DLGT= 76
LANGUAGE(S)..... ASSEMBLER
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH

02 SKILID                DISPLAY                  9(4)          1      4
02 SKILNAME              DISPLAY                  X(12)          5     12
02 SKILDESC              DISPLAY                  X(60)         17     60

*****

REC SYNONYM NAME... SKILF                          SKILL                      RLGT= 96
REC SYNONYM VER.... 0100                             DLGT= 76
LANGUAGE(S)..... FORTRAN
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH

02 SKLID                DISPLAY                  9(4)          1      4
02 SKLNAME              DISPLAY                  X(12)          5     12
02 SKLDSC              DISPLAY                  X(60)         17     60
IDMSRPTS 14.1          -- SCHEMA RECORD DESCRIPTION LISTING ---      DATE      TIME      PAGE
RECDES                DICTIONARY DEFAULT OF NODE DEFAULT      02/13/01  142430    28
                  SCHEMA EMPSCHM VERSION 100

RECORD NAME..... STRUCTURE                          RLGT= 36
RECORD VERSION..... 0100                             DLGT= 12
RECORD ID..... 0460                                KLGT= 24
RECORD LENGTH..... FIXED                             DSTRT= 24
LOCATION MODE..... VIA SET   MANAGES                  DISPLACEMENT 0000 PAGES
WITHIN..... EMP-DEMO-REGION   OFFSET               5 PGS FOR      95 PGS
DBKEY POSITIONS... SET..... TYPE..... NEXT   PRIOR OWNER
                  MANAGES      MEMBER          1      2      3
                  REPORTS-TO    MEMBER          4      5      6
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT LGTH

02 STRUCTURE-CODE-0460    DISPLAY                  X(2)          1      2

```

```
88 ADMIN-0460          COND          'A'                                1
88 PROJECT-0460        COND                                1
                        'P1'                                THRU 'P9'

02 STRUCTURE-DATE-0460 DISPLAY                                3      8
03 STRUCTURE-YEAR-0460 DISPLAY                                3      4
                        9(4)
03 STRUCTURE-MONTH-0460 DISPLAY                                7      2
                        9(2)
03 STRUCTURE-DAY-0460  DISPLAY                                9      2
                        9(2)
02 FILLER              DISPLAY                                11     2
                        XX

*****

REC SYNONYM NAME... STRUCTUR          STRUCTURE          RLGTH=   36
REC SYNONYM VER.... 0100              DLGTH=   12
LANGUAGE(S)..... ASSEMBLER
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT  LGTH

02 STRCODE              DISPLAY          X(2)              1      2
88 ADMIN                COND            'A'                1
88 PROJECT              COND                                1
                        'P1'                                THRU 'P9'

02 STRDATE              DISPLAY                                3      8
03 STRYEAR              DISPLAY          9(4)              3      4
03 STRMONTH             DISPLAY          9(2)              7      2
03 STRDAY               DISPLAY          9(2)              9      2
02 FILLER               DISPLAY          XX                11     2

*****

REC SYNONYM NAME... STRUCT          STRUCTURE          RLGTH=   36
REC SYNONYM VER.... 0100              DLGTH=   12
LANGUAGE(S)..... FORTRAN
DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT  LGTH

IDMSRPTS 14.1          -- SCHEMA RECORD DESCRIPTION LISTING ---      DATE      TIME      PAGE
RECDES                DICTIONARY DEFAULT OF NODE DEFAULT          02/13/01 142430   29
                        SCHEMA EMPSCHM VERSION 100

DATA ITEM..... REDEFINES..... USAGE..... VALUE..... PICTURE..... STRT  LGTH

02 STCODE              DISPLAY          X(2)              1      2
88 ADMIN                COND            'A'                1
88 PROJCT              COND                                1
                        'P1'                                THRU 'P9'

02 STDATE              DISPLAY                                3      8
03 STYEAR              DISPLAY          9(4)              3      4
03 STMNTN             DISPLAY          9(2)              7      2
03 STDAY               DISPLAY          9(2)              9      2
02 FILLER              DISPLAY          XX                11     2
IDMSRPTS 14.1          -- SCHEMA RECORD DESCRIPTION LISTING ---      DATE      TIME      PAGE
RECDES                DICTIONARY DEFAULT OF NODE DEFAULT          02/13/01 142430   30
                        SCHEMA EMPSCHM VERSION 100
```

IDMSRPTS 14.1

----- END OF REPORT -----
----- INPUT PARAMETER LISTING -----
DICTIONARY DEFAULT OF NODE DEFAULT

DATE	TIME	PAGE
02/13/01	142430	1

END OF INPUT PARAMETERS

%%

APPENDIX C

Tracing CA-IDMS Data Adapter Processing

Topics:

- Using Traces
- Allocating Traces
- Activating, Deactivating, and Querying Traces
- Sample Traces

The data adapter communicates with the CA-IDMS/DB DBMS through DML calls it generates on your behalf. You can view the outcome of each call using the data adapter FSTRACE facilities. Traces are helpful for debugging and performance analysis. The data adapter traces are easy to invoke, require no changes to either the data adapter or request, and have no effect on how the data adapter functions.

Using Traces

Each trace is associated with a trace level and a default ddname. Using a FOCUS SET command, you must turn on each trace level you want to generate prior to issuing the request to be traced. When turning a trace on, you can also change its associated ddname. You must make sure that the relevant ddnames are allocated prior to activating each trace.

The following trace levels are available for the CA-IDMS Data Adapter:

- Level 1 (default ddname FSTRACE) shows each call to IDMS and its result.
- Level 2 (default ddname FSTRACEP) shows each call to IDMS and, additionally, all control blocks and parameters passed to and from IDMS.

Allocating Traces

You can allocate all or any combination of traces during your session or in batch. You can display the results online or store them in a file or sequential data set.

Tip:

The trace facilities are intended for use in query optimization and problem debugging. Application programs should not be written to depend on the format or content of any trace, as they may change in later releases.

Syntax

How to Allocate a Trace Online

You can store trace information in an MVS sequential data set or CMS file, or you can display it online at the terminal. To capture trace data, issue the appropriate command from the command level. For example, to allocate ddname FSTRACE, issue one of the following commands:

```
{MVS|TSO} ALLOC F(ddname) DA('userid.FSTRACE') SHR REU
```

or

```
DYNAM ALLOC DD FSTRACE DATASET userid.FSTRACE SHR REUSE
```

or:

```
CMS FILEDEF FSTRACE DISK FSTRACE DATA A
```

Note:

DCB attributes are LRECL=80 and RECFM=F.

To view the trace information, use the system editor or the FOCUS TED editor.

To display trace data at the terminal, issue the appropriate command from the command level

```
{MVS|TSO} ALLOC F(FSTRACE) DA(*)
```

or

```
DYNAM ALLOC FILE FSTRACE DA *
```

or:

```
CMS FILEDEF FSTRACE TERMINAL
```

Syntax**How to Allocate a Trace in Batch**

You can write trace results to SYSOUT. BLKSIZE information is optional, but should be compatible with other FSTRACE formats. For example, to allocate ddname FSTRACE:

```
//FSTRACE DD SYSOUT=*,DCB=(LRECL=80,BLKSIZE=80,RECFM=F)
```

You can also write trace results to an MVS sequential data set. First, allocate the FSTRACE data set in a prior batch step (as shown) or in ISPF:

```
//ALLOC EXEC PGM=IEFBR14
//FSTRACE DD DISP=(,CATLG),DSN=userid.FSTRACE,
// UNIT=SYSDA,VOL=SER=USERM1,SPACE=(TRK,(5,5)),
// DCB=(LRECL=80,BLKSIZE=80,RECFM=F)
.
.
.
```

Then, allocate the trace data set with DISP=MOD in the batch FOCUS JCL:

```
.
.
.
//FOCBATCH EXEC PGM=FOCUS
//FSTRACE DD DISP=(MOD,KEEP,KEEP),DSN=userid.FSTRACE
```

Syntax**How to Free Trace Allocations**

To disable a data adapter trace, clear the associated allocation

```
{MVS|TSO} FREE F(ddname)
```

or

```
DYNAM FREE FILE ddname
```

or

```
CMS FILEDEF ddname CLEAR
```

where:

ddname

Is the ddname associated with the trace, by default either FSTRACE or FSTRACEP.

Activating, Deactivating, and Querying Traces

Each trace is assigned a level and must be associated with a component name. For the CA-IDMS Data Adapter, the component name is IDMSR.

Syntax

How to Activate CA-IDMS Data Adapter Traces

You can activate both trace levels or one specific trace level with the SET TRACEON command. You can issue this request at the command level, in a FOCEXEC, or in any supported profile. You must issue this command prior to running the request for which you want the trace generated. You can also identify a ddname to associate with the trace. The syntax is one of the following

```
SET TRACEON = IDMSR [ / level ]  
SET TRACEON = IDMSR / [ level ] [ / [ ddname ] ]
```

where:

IDMSR

Indicates that the command applies to traces for the CA-IDMS Data Adapter.

level

Is an integer that identifies the trace level to activate. Valid values are 1 and 2. If omitted, all trace levels are activated.

ddname

Is the ddname to be used. If omitted, any ddname previously associated with the trace level during the session is used. If no ddname has been associated with a trace level, the default ddname is used. The default ddname for level 1 is FSTRACE. The default ddname for level 2 is FSTRACEP. If both trace levels are activated, both types of trace information appear in the trace output.

Example

Activating the CA-IDMS Data Adapter Trace Facility

The following command turns on traces for all levels of the IDMSR component and uses their current ddnames. If no ddname has been specified, it uses the default ddnames:

```
SET TRACEON = IDMSR
```

The following example turns on the level 2 trace and associates it with the current ddname. If no ddname has been specified, it uses the default ddname, FSTRACEP:

```
SET TRACEON = IDMSR/2
```

The following example turns on the level 1 trace and associates it with ddname SYSPRINT:

```
SET TRACEON = IDMSR/1/SYSPRINT
```

Syntax

How to Deactivate CA-IDMS Data Adapter Traces

You can deactivate all trace levels or one specific trace level with the SET TRACEOFF command. The syntax is one of the following

```
SET TRACEOFF = [IDMSR] [ / level ]  
SET TRACEOFF = [IDMSR] [ // ddname ]
```


where:

IDMSR

Indicates that the command applies to traces for the CA-IDMS Data Adapter. If omitted, the command applies to all components.

level

Is an integer that identifies the trace level to be deactivated. Valid values are 1 and 2. If omitted, all trace levels are deactivated.

ddname

Turns off the trace levels associated with *ddname*.

Example

Deactivating CA-IDMS Data Adapter Traces

The following command turns off trace level 1 for the CA-IDMS Data Adapter:

```
SET TRACEOFF = IDMSR/1
```

The following command turns off all traces for the CA-IDMS Data Adapter:

```
SET TRACEOFF = IDMSR
```

The following command turns off all traces allocated to *ddname* FSTRACE for all components:

```
SET TRACEOFF = //FSTRACE
```

Syntax

How to Query Traces

The trace query commands tell you which trace levels are either activated or deactivated for *every* component. Be aware that these commands may show information that does not pertain to IDMS.

To list all of the trace level/component combinations currently active, issue the following command:

```
SET TRACEON = ?
```

To list all of the trace level/component combinations not currently active, issue the following command:

```
SET TRACEOFF = ?
```

Note:

Be aware that this command may produce a voluminous display listing information about components in which you have no interest.

Example

Querying Traces

```
SET TRACEON = IDMSR
```

```
> > SET TRACEON = ?
```

(FOC1180)	Name	Level	DDName	Description	Set	Filedef	Active
(FOC1180)	IDMSR	1	FSTRACE	IDMS The Calls Only	Y	Y	Y
(FOC1180)	IDMSR	2	FSTRACEP	IDMS The Parameters	Y	Y	Y

Sample Traces

This topic contains a sample trace for each trace level. Chapter 6, *Creating File Descriptions With AUTOIDMS*, contains a sample session that demonstrates how to create the EMPFILE Master and Access Files used in the examples.

Example

IDMSR Level 1 Trace

The following example allocates the trace a sequential file, turns on the level 1 trace, and runs a request:

```
DYNAM ALLOC DD FSTRACE  DA PMSSAE.FSTRACE SHR REU
SET TRACEON = IDMSR/1/FSTRACE
TABLE FILE EMPFILE
PRINT EMP_LAST_NAME AS 'LAST NAME'
EMP_FIRST_NAME AS 'FIRST NAME'
DEPT_NAME AS 'DEPARTMENT'
  IF EMP_LAST_NAME EQ 'CLOTH'
  IF EMP_FIRST_NAME EQ 'TERRY'
END
```

The following trace results are stored in the file. The level 1 trace shows each call to IDMS and the index or key values passed and retrieved:

```
*****
***** BIND RUN-UNIT *****
*****

*****
***** CALL COMPLETED *****
*****

*****
***** BIND RECORD *****
*****
>>>> EMPLOYEE

*****
***** CALL COMPLETED *****
*****
```

```
*****
***** BIND RECORD *****
*****
>>>> DEPARTMENT
```

```
*****
***** CALL COMPLETED *****
*****
```

```
*****
***** READY *****
*****
```

```
>>>> EMP-DEMO-REGION
```

```
*****
***** CALL COMPLETED *****
*****
```

```
*****
***** READY *****
*****
```

```
>>>> ORG-DEMO-REGION
```

```
*****
***** CALL COMPLETED *****
*****
```

INDEX VALUE:		
C5D90D8EDE70404C3D3D6E3C84040404040404040E3	*CLOTH	TERRY
INDEX LIMIT:		
C5D90D8932B0404C3D3D6E3C84040404040404040E3	*CLOTH	TERRY

```
*****
***** OBTAIN WITHIN SET USING VALUE ***
*****
>>>> EMPLOYEE
```

```
*****
***** CALL COMPLETED *****
*****
```

INDEX RETRIEVED:		
C5D90D8EDD70404C3D3D6E3C84040404040404040E3	*CLOTH	TERRY

```
*****
***** IF MEMBER *****
*****
>>>> DEPT-EMPLOYEE

*****
***** CALL COMPLETED *****
*****

*****
***** OBTAIN OWNER *****
*****
>>>> DEPT-EMPLOYEE

*****
***** CALL COMPLETED *****
*****

*****
***** OBTAIN NEXT MEMBER *****
*****
>>>> EMPLOYEE

*****
***** CALL COMPLETED *****
*****

INDEX RETRIEVED:
C5E30D8EDD70404C3D3D6E4C44040404040404040C2      *CLOUD      BETH
-----

*****
***** FINISH *****
*****

*****
***** CALL COMPLETED *****
*****
```

The output of the request is:

LAST NAME	FIRST NAME	DEPARTMENT
-----	-----	-----
CLOTH	TERRY	THERMOREGULATION

Example

IDMSR Level 2 Trace

The following example allocates the trace a sequential file, turns on the level 2 trace, and runs a request:

```
DYNAM ALLOC DD FSTRACEP  DA PMSSAE.FSTRACEP SHR REU
SET TRACEON = IDMSR/2/FSTRACEP
TABLE FILE EMPFILE
PRINT EMP_LAST_NAME AS 'LAST NAME'
EMP_FIRST_NAME AS 'FIRST NAME'
DEPT_NAME AS 'DEPARTMENT'
  IF EMP_LAST_NAME EQ 'CLOTH'
  IF EMP_FIRST_NAME EQ 'TERRY'
END
```

The following trace results are stored in the file. The level 2 trace shows each call to IDMS and the control blocks, parameters and key or index values passed and retrieved:

```
*****
***** BIND RUN-UNIT *****
*****

*****
**** PARAMETERS TO IDMS: *****
*****

SUBSCHEMA CONTROL BLOCK:
C5E30D889CD8404C6D6C3E4E2404040F1F4F0F000000000 *FOCUS 1400...ETH
C5E30D889CE8404404040404040404040404040404040 * ETH
C5E30D889CF8404404040404040404040404040404040 * ETH
C5E30D889D08404404040404040404040404040404040 * ETH
C5E30D889D18404404040404040404040404040404040 * ETH
C5E30D889D28404404040404040404040404040404040 * ETH
C5E30D889D38404404040404040404040404040404040 * ETH
C5E30D889D48404404040404040404040404040404040 * ETH
C5E30D889D58404404040404040404040404040404040 * ETH
C5E30D889D68404404040404040404040404040404040 * ETH
C5E30D889D78404404040404040404040404040404040 * ETH
C5E30D889D88404404040404040404040404040404040 * ETH
C5E30D889D984044040404000000000040404040404000 * ... .ETH
C5E30D889DA840400000000000000000000000000000000 *.. ETH

IDBMSCOM PARAMETER:
C5E30D889D72404404040400000000000000000000000000 * ETH

REC:
C5E30D889CD8404C6D6C3E4E2404040F1F4F0F000000000 *FOCUS 1400...ETH

*****
***** CALL COMPLETED *****
*****
```

Information Builders

```
C5E30D80CA3A404C5D4D7D3D6E8C5C540404040404040      *EMPLOYEE      ETH

*****
***** CALL COMPLETED *****
*****

*****
**** PARAMETERS FROM IDMS: ****
*****

SUBSCHEMA STATUS:
C5E30D889CE0404F0F0F0F0000000000000000000000000000      *0000      ETH

SUBSCHEMA CONTROL BLOCK:
C5E30D889CD8404C6D6C3E4E2404040F0F0F0F00000FF08      *FOCUS      0000...ETH
C5E30D889CE840440404040404040404040404040404040      *      ETH
C5E30D889CF840440404040404040404040404040404040      *      ETH
C5E30D889D0840440404040404040404040404040404040      *      ETH
C5E30D889D1840440404040404040404040404040404040      *      ETH
C5E30D889D2840440404040404040404040404040404040      *      ETH
C5E30D889D3840440404040404040404040404040404040      *      ETH
C5E30D889D4840440404040404040404040404040404040      *      ETH
C5E30D889D5840440404040404040404040404040404040      *      ETH
C5E30D889D6840440404040000CF000D9D7C24040404040      *      ..0.RPB      ETH
C5E30D889D78404404040404040404040404040E2E2C3C1      *      SSAETH
C5E30D889D884040D91600840404040404040E5C9C2C1      *.j..      VIBAETH
C5E30D889D98404FFFFFFFFFF0000000040404040404000      *......      .ETH
C5E30D889DA840400000000000000000000000000000000      *..      ETH

*****
***** BIND RECORD *****
*****

>>>> DEPARTMENT

*****
**** PARAMETERS TO IDMS: ****
*****

SUBSCHEMA CONTROL BLOCK:
C5E30D889CD8404C6D6C3E4E2404040F0F0F0F00000FF08      *FOCUS      0000...ETH
C5E30D889CE840440404040404040404040404040404040      *      ETH
C5E30D889CF840440404040404040404040404040404040      *      ETH
C5E30D889D0840440404040404040404040404040404040      *      ETH
C5E30D889D1840440404040404040404040404040404040      *      ETH
C5E30D889D2840440404040404040404040404040404040      *      ETH
C5E30D889D3840440404040404040404040404040404040      *      ETH
C5E30D889D4840440404040404040404040404040404040      *      ETH
C5E30D889D5840440404040404040404040404040404040      *      ETH
C5E30D889D6840440404040000CF000D9D7C24040404040      *      ..0.RPB      ETH
C5E30D889D784044040404040404040404040E2E2C3C1      *      SSAETH
C5E30D889D884040D91600840404040404040E5C9C2C1      *.j..      VIBAETH
C5E30D889D98404FFFFFFFFFF0000000040404040404000      *......      .ETH
C5E30D889DA840400000000000000000000000000000000      *..      ETH
```

Information Builders


```

SUBSCHEMA CONTROL BLOCK:
C5E30D889CD8404C6D6C3E4E2404040F0F0F0F00000FF08 *FOCUS 0000...ETH
C5E30D889CE840440404040404040404040404040404040 * ETH
C5E30D889CF840440404040404040404040404040404040 * ETH
C5E30D889D0840440404040404040404040404040404040 * ETH
C5E30D889D1840440404040404040404040404040404040 * ETH
C5E30D889D2840440404040404040404040404040404040 * ETH
C5E30D889D3840440404040404040404040404040404040 * ETH
C5E30D889D4840440404040404040404040404040404040 * ETH
C5E30D889D5840440404040404040404040404040404040 * ETH
C5E30D889D684044040404040000CF000D9D7C24040404040 * ..0.RPB ETH
C5E30D889D784044040404040404040404040E2E2C3C1 * SSAETH
C5E30D889D884040D91600840404040404040E5C9C2C1 *.j.. VIBAETH
C5E30D889D98404FFFFFFFFFF00000000404040404040000 *..... .ETH
C5E30D889DA8404000000000000000000000000000000000 *.. ETH

IDBMSCOM PARAMETER:
C5E30D889D5C404404040400000000000000000000000000 * ETH

REC:
C5E30D80C340404C5D4D760C4C5D4D660D9C5C7C9D6D540 *EMP.DEMO.REGION ETH

*****
***** CALL COMPLETED *****
*****

*****
***** PARAMETERS FROM IDMS: *****
*****

SUBSCHEMA STATUS:
C5E30D889CE0404F0F0F0F0000000000000000000000000000 *0000 ETH

SUBSCHEMA CONTROL BLOCK:
C5E30D889CD8404C6D6C3E4E2404040F0F0F0F00000FF08 *FOCUS 0000...ETH
C5E30D889CE840440404040404040404040404040404040 * ETH
C5E30D889CF8404C5D4D760C4C5D4D660D9C5C7C9D6D540 *EMP.DEMO.REGION ETH
C5E30D889D0840440404040404040404040404040404040 * ETH
C5E30D889D1840440404040404040404040404040404040 * ETH
C5E30D889D2840440404040404040404040404040404040 * ETH
C5E30D889D3840440404040404040404040404040404040 * ETH
C5E30D889D4840440404040404040404040404040404040 * ETH
C5E30D889D5840440404040404040404040404040404040 * ETH
C5E30D889D684044040404040000CF000D9D7C24040404040 * ..0.RPB ETH
C5E30D889D784044040404040404040404040E2E2C3C1 * SSAETH
C5E30D889D884040D91600840404040404040E5C9C2C1 *.j.. VIBAETH
C5E30D889D98404FFFFFFFFFF00000000404040404040000 *..... .ETH
C5E30D889DA8404000000000000000000000000000000000 *.. ETH

*****
***** READY *****
*****

```

>>>> ORG-DEMO-REGION

**** PARAMETERS TO IDMS: *****

SUBSCHEMA CONTROL BLOCK:

C5E30D889CD8404C6D6C3E4E2404040F0F0F0F00000FF08	*FOCUS 0000...ETH
C5E30D889CE8404404040404040404040404040404040	* ETH
C5E30D889CF8404C5D4D760C4C5D4D660D9C5C7C9D6D540	*EMP.DEMO.REGION ETH
C5E30D889D08404404040404040404040404040404040	* ETH
C5E30D889D18404404040404040404040404040404040	* ETH
C5E30D889D28404404040404040404040404040404040	* ETH
C5E30D889D38404404040404040404040404040404040	* ETH
C5E30D889D48404404040404040404040404040404040	* ETH
C5E30D889D58404404040404040404040404040404040	* ETH
C5E30D889D6840440404040000CF000D9D7C24040404040	* ..0.RPB ETH
C5E30D889D78404404040404040404040404040E2E2C3C1	* SSCAETH
C5E30D889D884040D91600840404040404040E5C9C2C1	*.j.. VIBAETH
C5E30D889D98404FFFFFFF0000000040404040404000	*..... .ETH
C5E30D889DA8404000000000000000000000000000000	*.. ETH

IDBMSCOM PARAMETER:

C5E30D889D5C40440404040000000000000000000000	* ETH
--	-----------------

REC:

C5E30D80C368404D6D9C760C4C5D4D660D9C5C7C9D6D540	*ORG.DEMO.REGION ETH
---	----------------------

***** CALL COMPLETED *****

**** PARAMETERS FROM IDMS: *****

SUBSCHEMA STATUS:

C5E30D889CE0404F0F0F0F00000000000000000000000	*0000 ETH
---	---------------------

SUBSCHEMA CONTROL BLOCK:

C5E30D889CD8404C6D6C3E4E2404040F0F0F0F00000FF08	*FOCUS 0000...ETH
C5E30D889CE8404404040404040404040404040404040	* ETH
C5E30D889CF8404C5D4D760C4C5D4D660D9C5C7C9D6D540	*EMP.DEMO.REGION ETH
C5E30D889D08404404040404040404040404040404040	* ETH
C5E30D889D18404404040404040404040404040404040	* ETH
C5E30D889D28404404040404040404040404040404040	* ETH
C5E30D889D38404404040404040404040404040404040	* ETH
C5E30D889D48404404040404040404040404040404040	* ETH
C5E30D889D58404404040404040404040404040404040	* ETH
C5E30D889D6840440404040000CF000D9D7C24040404040	* ..0.RPB ETH
C5E30D889D784044040404040404040404040E2E2C3C1	* SSCAETH
C5E30D889D884040D91600840404040404040E5C9C2C1	*.j.. VIBAETH
C5E30D889D98404FFFFFFF0000000040404040404000	*..... .ETH
C5E30D889DA8404000000000000000000000000000000	*.. ETH

```

INDEX VALUE:
C5D90D8E3E70404C3D3D6E3C84040404040404040E3      *CLOTH          TERRY
INDEX LIMIT:
C5D90D889078404C3D3D6E3C8404040404040404040E3      *CLOTH          TERRY

*****
***** OBTAIN WITHIN SET USING VALUE *****
*****
>>>> EMPLOYEE

*****
***** PARAMETERS TO IDMS: *****
*****

SUBSCHEMA CONTROL BLOCK:
C5D90D889CD8404C6D6C3E4E240404F0F0F0F00000FF08      *FOCUS    0000...ERRY
C5D90D889CE8404404040404040404040404040404040      *              ERRY
C5D90D889CF8404C5D4D760C4C5D4D660D9C5C7C9D6D540      *EMP.DEMO.REGION ERRY
C5D90D889D08404404040404040404040404040404040      *              ERRY
C5D90D889D18404404040404040404040404040404040      *              ERRY
C5D90D889D28404404040404040404040404040404040      *              ERRY
C5D90D889D38404404040404040404040404040404040      *              ERRY
C5D90D889D48404404040404040404040404040404040      *              ERRY
C5D90D889D58404404040404040404040404040404040      *              ERRY
C5D90D889D6840440404040000CF000D9D7C24040404040      *      .0.RPB    ERRY
C5D90D889D7840440404040404040404040404040E2E2C3C1      *              SSAERRY
C5D90D889D884040D9160084040404040404040E5C9C2C1      *.j..          VIBAERRY
C5D90D889D98404FFFFFFFFFF0000000040404040404000      *......      .ERRY
C5D90D889DA840400000000000000000000000000000000      *.          ERRY

IDBMSCOM PARAMETER:
C5D90D889D58404404040400000000000000000000000000      *              ERRY

REC:
C5D90D80CA3A404C5D4D7D3D6E8C5C540404040404040      *EMPLOYEE      ERRY

*****
***** CALL COMPLETED *****
*****

*****
***** PARAMETERS FROM IDMS: *****
*****

```

Information Builders

```

REC:
C5D90D80CB6E404C4C5D7E360C5D4D7D3D6E8C5C5404040      *DEPT.EMPLOYEE      ERRY

*****
*****  CALL COMPLETED  *****
*****

*****
****  PARAMETERS FROM IDMS:  ****
*****

SUBSCHEMA STATUS:
C5D90D889CE0404F0F0F0F00000000000000000000000000000000      *0000      ERRY

SUBSCHEMA CONTROL BLOCK:
C5D90D889CD8404C6D6C3E4E2404040F0F0F0F001250701      *FOCUS      0000...ERRY
C5D90D889CE8404C5D4D7D3D6E8C5C540404040404040404040      *EMPLOYEE      ERRY
C5D90D889CF8404C5D4D760C4C5D4D660D9C5C7C9D6D540      *EMP.DEMO.REGION  ERRY
C5D90D889D084044040404040404040404040404040404040      *      ERRY
C5D90D889D184044040404040404040404040404040404040      *      ERRY
C5D90D889D284044040404040404040404040404040404040      *      ERRY
C5D90D889D384044040404040404040404040404040404040      *      ERRY
C5D90D889D484044040404040404040404040404040404040      *      ERRY
C5D90D889D584044040404040404040404040404040404040      *      ERRY
C5D90D889D6840440404040000CF000D9D7C2404040404040      *      ..0.RPB      ERRY
C5D90D889D784044040404040404040404040E2E2C3C1      *      SSSAERRY
C5D90D889D884040D9160084040404040404040E5C9C2C1      *.j..      VIBAERRY
C5D90D889D98404FFFFFFFFFF0000000040404040404000      *......      .ERRY
C5D90D889DA84040000000000000000000000000000000000      *.      ERRY

*****
*****  OBTAIN OWNER  *****
*****

>>>>  DEPT-EMPLOYEE

*****
****  PARAMETERS TO IDMS:  ****
*****

```

```

SUBSCHEMA CONTROL BLOCK:
C5D90D889CD8404C6D6C3E4E2404040F0F0F0F001250701 *FOCUS 0000...ERRY
C5D90D889CE8404C5D4D7D3D6E8C5C54040404040404040 *EMPLOYEE ERRY
C5D90D889CF8404C5D4D760C4C5D4D660D9C5C7C9D6D540 *EMP.DEMO.REGION ERRY
C5D90D889D0840440404040404040404040404040404040 * ERRY
C5D90D889D1840440404040404040404040404040404040 * ERRY
C5D90D889D2840440404040404040404040404040404040 * ERRY
C5D90D889D3840440404040404040404040404040404040 * ERRY
C5D90D889D4840440404040404040404040404040404040 * ERRY
C5D90D889D5840440404040404040404040404040404040 * ERRY
C5D90D889D6840440404040000CF000D9D7C24040404040 * ..0.RPB ERRY
C5D90D889D784044040404040404040404040E2E2C3C1 * SSAERRY
C5D90D889D884040D91600840404040404040E5C9C2C1 *.j.. VIBAERRY
C5D90D889D98404FFFFFFFFFF0000000040404040404000 *.ERRY
C5D90D889DA840400000000000000000000000000000000 *.ERRY

IDBMSCOM PARAMETER:
C5D90D889D5640440404040000000000000000000000000 * ERRY

SET:
C5D90D80CB6E404C4C5D7E360C5D4D7D3D6E8C5C5404040 *DEPT.EMPLOYEE ERRY

*****
***** CALL COMPLETED *****
*****

*****
***** PARAMETERS FROM IDMS: *****
*****

SUBSCHEMA STATUS:
C5D90D889CE0404F0F0F0F00000000000000000000000000 *0000 ERRY

SUBSCHEMA CONTROL BLOCK:
C5D90D889CD8404C6D6C3E4E2404040F0F0F0F00125A501 *FOCUS 0000..v.ERRY
C5D90D889CE8404C4C5D7C1D9E3D4C5D5E3404040404040 *DEPARTMENT ERRY
C5D90D889CF8404D6D9C760C4C5D4D660D9C5C7C9D6D540 *ORG.DEMO.REGION ERRY
C5D90D889D0840440404040404040404040404040404040 * ERRY
C5D90D889D1840440404040404040404040404040404040 * ERRY
C5D90D889D2840440404040404040404040404040404040 * ERRY
C5D90D889D3840440404040404040404040404040404040 * ERRY
C5D90D889D4840440404040404040404040404040404040 * ERRY
C5D90D889D5840440404040404040404040404040404040 * ERRY
C5D90D889D6840440404040000CF000D9D7C24040404040 * ..0.RPB ERRY
C5D90D889D784044040404040404040404040E2E2C3C1 * SSAERRY
C5D90D889D884040D91600840404040404040E5C9C2C1 *.j.. VIBAERRY
C5D90D889D98404FFFFFFFFFF0000000040404040404000 *.ERRY
C5D90D889DA840400000000000000000000000000000000 *.ERRY

```



```

SUBSCHEMA CONTROL BLOCK:
C5D90D889CD8404C6D6C3E4E2404040F0F0F0F001250101  *FOCUS    0000...ERRY
C5D90D889CE8404C5D4D7D3D6E8C5C54040404040404040  *EMPLOYEE      ERRY
C5D90D889CF8404C5D4D760C4C5D4D660D9C5C7C9D6D540  *EMP.DEMO.REGION ERRY
C5D90D889D0840440404040404040404040404040404040  *          ERRY
C5D90D889D1840440404040404040404040404040404040  *          ERRY
C5D90D889D2840440404040404040404040404040404040  *          ERRY
C5D90D889D3840440404040404040404040404040404040  *          ERRY
C5D90D889D4840440404040404040404040404040404040  *          ERRY
C5D90D889D5840440404040404040404040404040404040  *          ERRY
C5D90D889D6840440404040000CF000D9D7C24040404040  *      ..0.RPB  ERRY
C5D90D889D784044040404040404040404040E2E2C3C1  *          SSAERRY
C5D90D889D884040D91600840404040404040E5C9C2C1  *.j..        VIBAERRY
C5D90D889D98404FFFFFFFFFF0000000040404040404000  *. ....     .ERRY
C5D90D889DA840400000000000000000000000000000000  *..         ERRY

```

```

INDEX RETRIEVED:
C5E30D8E3D70404C3D3D6E4C4404040404040404040C2  *CLOUD      BETH

```

```

*****
***** FINISH *****
*****

```

```

*****
***** PARAMETERS TO IDMS: *****
*****

```

```

SUBSCHEMA CONTROL BLOCK:
C5E30D889CD8404C6D6C3E4E2404040F0F0F0F001250101  *FOCUS    0000...ETH
C5E30D889CE8404C5D4D7D3D6E8C5C54040404040404040  *EMPLOYEE      ETH
C5E30D889CF8404C5D4D760C4C5D4D660D9C5C7C9D6D540  *EMP.DEMO.REGION ETH
C5E30D889D0840440404040404040404040404040404040  *          ETH
C5E30D889D1840440404040404040404040404040404040  *          ETH
C5E30D889D2840440404040404040404040404040404040  *          ETH
C5E30D889D3840440404040404040404040404040404040  *          ETH
C5E30D889D4840440404040404040404040404040404040  *          ETH
C5E30D889D5840440404040404040404040404040404040  *          ETH
C5E30D889D6840440404040000CF000D9D7C24040404040  *      ..0.RPB  ETH
C5E30D889D784044040404040404040404040E2E2C3C1  *          SSAETH
C5E30D889D884040D916008404040404040E5C9C2C1  *.j..        VIBAETH
C5E30D889D98404FFFFFFFFFF0000000040404040404000  *. ....     .ETH
C5E30D889DA840400000000000000000000000000000000  *..         ETH

```

```

IDBMSCOM PARAMETER:
C5E30D889D394044040404000000000000000000000000  *          ETH

```

```

REC:
C5E30D889CD8404C6D6C3E4E2404040F0F0F0F001250101  *FOCUS    0000...ETH

```

```

*****
***** CALL COMPLETED *****

```



```
*****  
*****  
***** PARAMETERS FROM IDMS: *****  
*****  
  
SUBSCHEMA STATUS:  
C5E30D889CE040F0F0F0F000000000000000000000000    *0000    ETH  
  
SUBSCHEMA CONTROL BLOCK:  
C5E30D889CD8404C6D6C3E4E2404040F0F0F0F001250101   *FOCUS      0000...ETH  
C5E30D889CE840440404040404040404040404040404040   *              ETH  
C5E30D889CF8404C5D4D760C4C5D4D660D9C5C7C9D6D540   *EMP.DEMO.REGION ETH  
C5E30D889D084044040404040404040404040404040404   *              ETH  
C5E30D889D184044040404040404040404040404040404   *              ETH  
C5E30D889D284044040404040404040404040404040404   *              ETH  
C5E30D889D384044040404040404040404040404040404   *              ETH  
C5E30D889D484044040404040404040404040404040404   *              ETH  
C5E30D889D584044040404040404040404040404040404   *              ETH  
C5E30D889D6840440404040000CF000D9D7C240404040404   *          ..0.RPB  ETH  
C5E30D889D784044040404040404040404040404040E2E2C3C1 *              SSCAETH  
C5E30D889D884040000000040404040404040000000000   *....        ...ETH  
C5E30D889D984040000000000000000404040404040000   *. ....       .ETH  
C5E30D889DA8404000000000000000000000000000000000   *.           ETH
```

Glossary

ACCESS	Attribute that indicates the relationship that exists between record-types.
Access File	Is an extension of Master File syntax designed specifically for the peculiarities of the IDMS DBMS. The Access File provides information to map DML record-types and LRF-based records, including record and area names and, where needed, set or field name information.
ACCESS PDS	The data set name of the Access File PDS onto which the new member will be generated.
Access segment declarations	Indicate the IDMS record information and parent/descendant relationship for each network record-type or LRF record described as a segment in a Master File.
ALIAS	The ALIAS attribute specifies an alternative name for FIELDNAME. An alias may be to up 12 characters in length and must be unique.
ASF	Automatic System Facility. This facility generates subschemas and navigational paths based on the user's menu selection.
AUTOIDMS	Process that uses a series of Dialogue Manager FOCEXECs and standard TABLE requests to automatically create Master and Access Files for network record-types.
Bill-of-materials	Two record types linked by more than one set.
Common member	A record-type that is a member of two or more sets.
Common owner	A record-type that is the owner of two or more sets.
CRFILE	Attribute that is specified only for segments that are described remotely in another Master File.
CV	Central Version.
DBA	Database Administrator.

DBMS	Database Management System.
DBNAME	The DBNAME attribute supports multiple database access. It can be used in local or Central Version mode to translate the subschema name into the proper load modules for data access.
DDS	Distributed Database Systems.
DICTNODE	Supports secondary dictionaries in DDS environments.
DMCL	Device Media Control Language.
DML	Data Manipulation Language. DML access is the traditional method of IDMS database navigation. It is the network navigation facility.
Entity Relationship diagram	IDMS representation of record-types and set relationships within a database.
GETOWN	The GETOWN attribute allows or inhibits GET OWNER calls which obtain the owner records from a member record-type.
GROUP	The GROUP attribute identifies a set of fields following it with a single name. This GROUP name is any unique name up to 12 characters in length.
IDD	Integrated Data Dictionary.
IDMS	Integrated Database Management System.
Index-Based Retrieval	Using embedded cross-references, the field in the descendant segment represents an index on the IDMS record-type. The data adapter use the value from the parent segment and performs entry-level IDMS retrieval by “walking” the index set.
I/O	Input/Output.
Index declarations	Provide information about each IDMS index.
Integrated Index	A method of indexing where the database management system handles all index access and maintenance.
KL	The KL segment type is used when several Master Files are constructed for the same IDMS subschema. It is used if the underlying segment is non-unique (S).
KLU	The KLU segment type is used when several Master Files are constructed for the same IDMS subschema. It is used if underlying segment is unique (U).
Loop structures	Implement complicated relationships among record-types.

LR	Logical Records.
LRF	Logical Records Facility. LRF access is the “relational-like” method of access. LRF provides software to dynamically create “flat” records from one or more network record-types at run time.
MA	Mandatory/Automatic.
Master File	Contains attributes in comma-delimited format that define the field name for IDMS fields and segment relationships.
MASTER FILENAME	The name of the Master File to be generated; up to eight characters in length.
MASTER PDS	The data set name of the Master File PDS onto which the new member will be generated.
MEMBER NAME	The userid to be created on the Master and Access File partitioned data sets; up to eight characters in length.
MM	Mandatory/Manual.
Multi-member set	A set that contains more than one member record-type.
NODE	The NODE attribute supports IDMS Distributed Database Systems. The value for NODE is the IDMS data dictionary table entry that identifies the DDS node location of an IDMS distributed database.
OA	Optional/Automatic.
OM	Optional/Manual.
ORDER	The ORDER attribute creates a fictitious count field that may be decoded.
PDS	Partitioned data sets.
POSITION	The POSITION attribute directs FOCUS to OCCURS segments when non-repeating fields exist between repeating fields.
READY	Specifies when an LRF record is built from two or more physical record-types located in several databases.
SEQFIELD parameter	A method of index retrieval that does not require selection criteria and yet prohibits area sweeps on entry segments.
SEGNAME	The segment name may be a maximum of eight characters and must be unique within a given Master File.

SEGTYPE	Attribute that indicates whether a segment occurs once or many times.
SET	Implements a one-to-many relationship between record-types. The FOCUS equivalent of a set is the parent/descendant relationship between segments.
SETMBR	The SETMBR attribute specifies whether set membership is mandatory/automatic, mandatory/manual, optional/automatic, or optional/manual.
Simple set	A set relationship that corresponds to a parent/descendant relationship.
Subschema declarations	Identify the subschema used, the IDMS release under which the subschema was compiled, the calling mode to be used to retrieve records, and whether a trace is produced.
SUBSCHEMA NAME	IDMS subschema used to create the Master and Access Files; up to eight characters in length.
TED	System editor.
TRACE	Specifies whether a basic trace of all IDMS calls or a detailed list of all the parameters passed to IDMS will be displayed. The default is set for no trace.
TSO	Time Sharing Option.
UCF	Universal Communications Facility.
XMI	Cross-Machine Interface.

Index

Special Characters

? JOIN, 7-16

A

ACCESS attribute, 5-6, 5-7

Access Files, 5-1, 5-2

EMPFIL, B-34

EMPFULL, B-31

example, 5-3

index declarations, 5-2

LRF sample, B-21

network sample, B-19

segment declarations, 5-2

subschema declarations, 5-2

syntax, 5-2

accessing IDMS, 2-1

CMS environment, 1-2

Master and Access File requirements, 2-6

module IDMSR, 2-6

MVS environment, 1-2

accessing IDMS under CMS, 2-6

issuing FILEDEF commands, 2-6

accessing IDMS under MVS, 2-2

batch access, 2-4, 2-5

central version, 2-2

central version and local mode, 2-2

CLIST (central version), 2-2

CLIST (local mode), 2-3

IDMS and IDMSINTB load modules, 2-2

local mode, 2-3

MSO, 2-4

ACTUAL attribute, 4-6

ALIAS attribute, 4-5

ALL parameter, 7-9

ALL. prefix, 7-11

allocation of traces, C-2

alternate view, 7-12

AREA attribute, 5-5

area sweeps, 8-6

suppressing, 5-8

attributes, 4-2

ACCESS, 5-6, 5-7

ACTUAL, 4-6

ALIAS, 4-5

AREA, 5-5

CLCFLD, 5-5

CRFILE, 4-4

DBNAME, 5-3, 5-4

DICTNAME, 5-3, 5-4

DICTNODE, 5-3, 5-4

field, 4-5

FIELDNAME, 4-5

FILE, 4-3

for CALC record-type, 5-5

for CALC-based segments, 5-7

for index-based segments, 5-7

for Integrated Indexes, 5-9

for set-based segments, 5-6

GETOWN, 5-6

GROUP, 4-5

INDEXAREA, 5-3

IXAREA, 5-9

IXDUP, 5-9

IXFLD, 5-9

IXORD, 5-9

IXSET, 5-9

KEYFLD, 5-6, 5-7, 8-7

Master File, 4-2

MODE, 5-3

MULTMBR, 5-6, 8-7

NODE, 5-3, 5-4

OCCURS, 4-4

ORDER, 4-10

PARENT, 4-4

POSITION, 4-4, 4-10, 4-12

READY, 5-3, 5-4

RECORD, 5-5

RELEASE, 5-3

segment, 4-4, 5-5

attributes (*continued*)

- SEGNAM, 5-5
- SEGTYPE, 4-4
- SETDUP, 5-6, 5-7
- SETMBR, 5-6
- SETNAME, 5-6, 5-7
- SETORD, 5-6
- SSHEMA, 5-3
- SUFFIX, 4-3
- TRACE, 5-3, 5-4
- USAGE, 4-6

AUTOIDMS, 6-1

- editing generated files, 6-15
- EMPFIL Master and Access Files, 6-29
- EMPFULL Master and Access Files, 6-18
- field attributes, 6-13
- file and segment attributes, 6-12
- generated descriptions, 6-12
- main menu, 6-3
- main menu PF keys, 6-6
- memory requirements, 6-2
- MFDLIST, 6-15
- MVS execution, 6-2
- parameter log file, 6-16
- prerequisites, 6-2
- record of generated Master Files, 6-15
- record/set selection screen, 6-10
- record/set selection screen PF keys, 6-11
- restarting, 6-17
- root record selection screen, 6-8
- root record selection screen PF keys, 6-9
- sample sessions, 6-18
- special considerations, 6-17
- starting, 6-2
- VM/CMS execution, 6-2

B

batch, 2-4, 2-5

- access, 2-4, 2-5
- trace allocation, C-3

bill-of-materials, 3-10

- relationships, 3-9

C

CALC, 5-5

- field retrieval, 8-3
- record-type attributes, 5-5

CALC-based, 5-7

- record-type example, 5-8
- relationships, 3-13
- retrieval, 8-8
- segment attributes, 5-7

central version, 2-2

- CLIST, 2-2
- JCL, 2-4

CHECK FILE command, 7-4

CLCFLD attribute, 5-5

CLIST, 2-2

- for central version access, 2-2
- for local mode access, 2-3

CMS, 2-1

- access, 2-6
- Master and Access files, 4-2

COBOL-to-FOCUS conversion table, 4-6

common member relationships, 3-7

common owner relationships, 3-6

COUNT command and retrieval, 7-5

CRFILE attribute, 4-4

cross-environment access, 1-3

- Cross Machine Interface, 1-4
- Distributed Database System, 1-3
- Universal Communications Facility, 1-3

D

database key in Master File, 4-8

DBKEY, 8-2

- calculating a value, 8-3

DBNAME, 5-3
 attribute, 5-4
 example, 5-10
 SET parameter, 5-9

descendant segment retrieval, 8-6

DICTNAME, 5-3
 attribute, 5-3, 5-4
 example, 5-10
 SET parameter, 5-9

DICTNODE attribute, 5-3, 5-4

dummy field example, 4-7

DYNAM for allocating FSTRACE, C-2

E

efficiency, 1-2

EMPFILE Access File, B-34

EMPFILE Master and Access Files, 6-29
 AUTOIDMS main menu, 6-29
 AUTOIDMS record/set selection screen, 6-31
 AUTOIDMS root record selection screen, 6-31
 generating with AUTOIDMS, 6-32

EMPFILE Master File, B-33

EMPFULL Access File, B-31

EMPFULL Master and Access Files, 6-18
 AUTOIDMS main menu, 6-18
 AUTOIDMS record/set selection screen, 6-20
 AUTOIDMS root record selection screen, 6-19
 editing in AUTOIDMS, 6-25, 6-26
 generating with AUTOIDMS, 6-24
 viewing structure in AUTOIDMS, 6-26

EMPFULL Master File, B-22

EMPSCHM schema sample, B-2

EMPSS01 subschema sample, B-11

Entity Relationship diagram, 3-4

entry segment retrieval, 8-2

error messages, A-1
 and response codes, A-2
 errors files, A-2

generating, A-2

errors, A-1
 correctable by database administration, A-3
 correctable by operations, A-4

examples, 4-7
 Access File, 5-3
 activating traces, C-4
 CALC-based record-type, 5-8
 deactivating traces, C-5
 dummy field, 4-7
 filler field, 4-8
 group field, 4-7
 JOIN command, 7-17
 multiple IF tests, 7-7
 OCCURS segment, 4-11
 ORDER field, 4-14
 querying traces, C-5
 remote segment description, 4-9
 screening on descendant segments, 7-6
 screening on parent segments, 7-5
 SET ALL = OFF, 7-9
 SET ALL = ON, 7-10
 SET ALL = ON with screening conditions, 7-10
 setting DBNAME and DICTNAME, 5-10

F

field attributes, 4-5

FIELDNAME attribute, 4-5

FILE attribute, 4-3

file inversion, 7-12
 and short paths, 7-13

filler field example, 4-8

freeing traces, C-3

FSTRACE, C-2

G

GETOWN attribute, 5-6

GROUP attribute, 4-5

group field example, 4-7

I

- IDMSR module, 2-6
- IDMSR query command, 5-10
- index declarations, 5-9
- INDEXAREA attribute, 5-3
- index-based retrieval, 8-8
- index-based segment attributes, 5-7
- indexed retrieval, 8-4
 - SEQFIELD parameter, 8-5
- Integrated Index attributes, 5-9
- intra-record structures, 4-9
- IXAREA attribute, 5-9
- IXDUP attribute, 5-9
- IXFLD attribute, 5-9
- IXORD attribute, 5-9
- IXSET attribute, 5-9

J

- JCL, 2-4, 2-5
 - for MSO, 2-4
 - trace allocation, C-3
- JOIN CLEAR command, 7-16
- JOIN command, 7-14
 - clearing, 7-16
 - example, 7-17
 - query, 7-16
 - syntax, 7-14
 - usage notes, 7-16

K

- keyed retrieval, 8-2
- KEYFLD attribute, 5-6, 5-7
 - and retrieval, 8-7

L

- levels for traces, C-2
- local mode, 2-3
 - CLIST, 2-3
 - JCL, 2-5
- loop structure relationships, 3-12
- LRF, 3-2
 - command formats, 8-9
 - record retrieval, 8-9
 - record sample, B-21
 - subschema sample, B-20

M

- main menu, 6-3
- mapping record-types to a segment, 3-2
- Master Files, 4-1
 - attributes, 4-2
 - EMPFIL, B-33
 - EMPFULL, B-22
 - IDMS database key, 4-8
 - in CMS, 4-2
 - in MVS, 4-2
 - LRF sample, B-20
 - network sample, B-14
- memory requirements, 1-4
- MFDLIST, 6-15
- MODE attribute, 5-3
- MSO, 2-1
 - JCL, 2-4
- multi-member relationships, 3-8
- multi-tiered relationships, 3-10
- MULTMBR attribute, 5-6
 - and retrieval, 8-7
- MVS, 2-1
 - batch access, 2-4, 2-5
 - interactive access in MSO, 2-4
 - Master and Access files, 4-2

N

network and LRF samples, B-1
network concepts, 3-2
network relationships, 3-14
NODE attribute, 5-3, 5-4

O

OCCURS attribute, 4-4
OCCURS segment, 4-9
 example, 4-11
 restrictions, 4-11
operating systems, 1-2
operation modes, 1-2
 IDMSOPTI, 1-2
 SYSSCTL DDname, 1-2
ORDER attribute, 4-10
ORDER field, 4-13
 example, 4-14
 usage rules, 4-13

P

parameter log file, 6-16
PARENT attribute, 4-4
parent/descendant relationships, 3-4
 common member, 3-7
 simple set, 3-4
PF key functions, 6-6
 AUTOIDMS main menu, 6-6
 AUTOIDMS record/set selection screen, 6-11
 AUTOIDMS root record selection screen, 6-9
POSITION attribute, 4-4, 4-10, 4-12

Q

query commands, 5-10
 ? JOIN, 7-16
 SET, 5-10

R

READLIMIT test, 7-7
READY attribute, 5-3, 5-4
RECORD attribute, 5-5
record/set selection screen, 6-10
RECORDLIMIT test, 7-7
record-types mapped to segments, 3-2
relationships, 3-3
 bill-of-materials, 3-9
 CALC-based, 3-13
 common member, 3-7
 common owner, 3-6
 loop structures, 3-12
 multi-member, 3-8
 set-based, 3-3
 simple set, 3-4
 summary, 3-14
RELEASE attribute, 5-3
remote file descriptions, 4-4
remote segment descriptions, 4-9
 example, 4-9
retrieval, 7-1
 and short paths, 7-8
 and sort phrases, 7-4
 of unique segments, 7-4
 record order, 7-3
 restricting the number of records retrieved, 7-7
 with screening conditions, 7-5
 with the COUNT command, 7-5
 with the SUM command, 7-5
retrieval methods, 8-1
 ACCESS parameter, 8-6
 area sweep, 8-6
 CALC field, 8-3
 CALC-based, 8-8
 descendent segments, 8-6
 DML network access, 3-2
 entry segment, 8-2
 IDMS database key, 8-2
 IDMS index, 8-4

- index-based, 8-8
- LRF records, 8-9
- LRF-based access, 3-2
- SEQFIELD parameter, 8-5
- set-based, 8-7

retrieval subtree, 7-2

root record selection screen, 6-8

S

samples, B-1

- CA-IDMS report, B-35
- EMPFIL Access File, B-34
- EMPFIL Master File, B-33
- EMPFULL Access File, B-31
- EMPFULL Master File, B-22
- EMPSCHM schema, B-2
- EMPSS01 subschema, B-11
- level 1 trace, C-6
- level 2 trace, C-9
- LRF Access File, B-21
- LRF Master File, B-20
- network Master File, B-14
- network Access File, B-19
- partial LRF record, B-21
- traces, C-6

schema sample, B-2

screening conditions, 7-5

- multiple IF tests, 7-7
- on descendant segments, 7-6
- on parent segments, 7-5
- with unique segments, 7-8

security features, 1-3

segment attributes, 4-4, 5-5

segments and record-types, 3-2

SEGNAM attribute, 5-5

SEGTYPE attribute, 4-4

SEQFIELD parameter and retrieval, 8-5

SET parameters, 5-10

- ALL, 7-9
- DBNAME, 5-10

- DICTNAME, 5-10
- TRACEON, C-4

set-based relationships, 3-3

- bill-of-materials, 3-9
- CALC- and indexed-based relationships, 3-13
- common member, 3-7
- common owner, 3-6
- Entity Relationship diagram, 3-4
- loop structures, 3-12
- multi-member, 3-8
- simple set, 3-4

set-based retrieval, 8-7

set-based segment attributes, 5-6

SETDUP attribute, 5-6, 5-7

SETMBR attribute, 5-6

SETNAME attribute, 5-6, 5-7

SETORD attribute, 5-6

short paths, 7-8

- and file inversion, 7-13
- in non-unique descendants, 7-9

simple set relationships, 3-4

sort phrases and retrieval, 7-4

SSHEMA attribute, 5-3

subschema declaration attributes, 5-3

subschema sample, B-11

subtree for retrieval, 7-2

SUFFIX attribute, 4-3

SUM command and retrieval, 7-5

suppressing area sweeps, 5-8

T

TRACE attribute, 5-3, 5-4

traces, C-1

- activating, C-4
- allocating, C-2
- batch mode, C-3

- deactivating, C-4
- disabling, C-3
- FSTRACE, C-2
- level 1 sample, C-6
- level 2 sample, C-9
- querying, C-5
- samples, C-6
- trace levels, C-2

- USAGE attribute, 4-6

- troubleshooting, A-1

U

- unique segments and screening conditions, 7-8

Reader Comments

In an ongoing effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please use this form to relay suggestions for improving this publication or to alert us to corrections. Identify specific pages where applicable. Send comments to:

Corporate Publications
Attn: Manager of Documentation Services
Information Builders
Two Penn Plaza
New York, NY 10121-2898

or FAX this page to (212) 967-0460, or call **Sara Elam** at (212) 736-4433, x**3207**.

Name: _____

Company: _____

Address: _____

Telephone: _____ Date: _____

Comments:

Reader Comments